NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INTEGRATION OF MULTIPLE UAVS FOR COLLABORATIVE ISR MISSIONS IN AN URBAN ENVIRONMENT

by

Chee Nam Chua

September 2012

Thesis Advisor:	Oleg Yakimenko
Thesis Co-Advisor:	Fotis Papoulias
Second Reader:	Gerard Leng

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2012	Master's Thesis

4. TITLE AND SUBTITLE Integration of Multiple UAVs for Collaborative ISR Missions in an Urban Environment

5. FUNDING NUMBERS

6. AUTHOR(S) Chee Nam Chua

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number N/A .

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Military conflicts are shifting from jungles and deserts to cities. This is because terrorists, insurgents, and guerillas find these areas provide a rich target environment and good hideouts. With the use of UAVs, urban threats can be tracked and targeted effectively. However, in an urban environment where there is little or no GPS signals and many obstacles, navigation of UAVs is a major challenge. Multiple UAVs can be employed to share sensor information to counter these challenges and to perform Intelligence, Surveillance, and Reconnaissance (ISR) missions with greater ground coverage and better success rates.

This thesis explored the various types of UAVs deployed for urban operations and investigated the trends of the UAVs in terms of their parameters such as weight, altitude, speed, and sensor suite. The challenges and requirements for interoperability of multi-UAV operations in urban environments were also discussed.

A direct-method-based control system for multiple UAV collaboration and obstacle collision avoidance was proposed. The UAVs were able to share and integrate their sensors' information for joint cooperation. A dynamic model was developed for the simulation testing of the algorithm. Following that, physical experiment was carried out in an indoor environment on Quanser QBall-X4 UAV to evaluate the results.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Quadrotor, UAVs, Interoperability, Direct method, Urban environment, Trajectory Following, Trajectory Generation, Optimization, Inverse Dynamics, IDVD	117
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

INTEGRATION OF MULTIPLE UAVS FOR COLLABORATIVE ISR MISSIONS IN AN URBAN ENVIRONMENT

Chee Nam Chua
Civilian Engineer, ST Aerospace Ltd.
B.S., Nanyang Technological University of Singapore, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2012

Author: Chee Nam Chua

Approved by: Oleg Yakimenko
Thesis Advisor

Fotis Papoulias
Thesis Co-Advisor

Gerard Leng
Second Reader

Clifford Whitcomb
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Military conflicts are shifting from jungles and deserts to cities. This is because terrorists, insurgents, and guerillas find these areas provide a rich target environment and good hideouts. With the use of UAVs, urban threats can be tracked and targeted effectively. However, in an urban environment where there is little or no GPS signals and many obstacles, navigation of UAVs is a major challenge. Multiple UAVs can be employed to share sensor information to counter these challenges and to perform Intelligence, Surveillance, and Reconnaissance (ISR) missions with greater ground coverage and better success rates.

This thesis explored the various types of UAVs deployed for urban operations and investigated the trends of the UAVs in terms of their parameters such as weight, altitude, speed, and sensor suite. The challenges and requirements for interoperability of multi-UAV operations in urban environments were also discussed.

A direct-method-based control system for multiple UAV collaboration and obstacle collision avoidance was proposed. The UAVs were able to share and integrate their sensors' information for joint cooperation. A dynamic model was developed for the simulation testing of the algorithm. Following that, physical experiment was carried out in an indoor environment on Quanser QBall-X4 UAV to evaluate the results.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

EFFBD	Enhanced functional flow block diagram
EMMI	Energy, Matter, Material and Information
EO/IR	Electro-optic / Infra-red
GCS	Ground control station
GPS	Global positioning systems
GNC	Guidance, navigation and control
IDVD	Inverse Dynamics in Virtual Domain
IED	Improvised explosive device
ISR	Intelligence, surveillance, and reconnaissance
LED	Light Emitting Diode
LOS	Line of sight
MTOW	Maximum Take-Off Weight
NED	North-East-Down
PL	Payload
ROE	Rules of engagement
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicle
UO	Urban operations
UTZ	Urban terrain zone
VTOL	Vertical take-off/land
UxVs	Unmanned Aerial, Surface and Ground Vehicles

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

This thesis explored the requirements of a collaborative ISR mission performed in an urban environment which poses severe challenges in communication and navigation with its complex and congested environment. Furthermore, various boundaries (political, social, economic, and physical) and stakeholders are involved in the operation of such a system. A concept of operation is presented with an emphasis on meeting the stakeholders' needs and discussing the key challenges that may be faced in the interoperability of such system.

In order to achieve higher autonomy with minimal user input to the system, a control approach is recommended in this thesis. Using the Inverse Dynamic in Virtual Domain (IDVD) method to generate quasi-optimal trajectory that allows real-time control for cooperation of multiple UxVs, the user just needs to key in the flight time, along with the initial and final points of the UAV, to fly. The algorithms will then generate a quasi-optimal feasible trajectory for the UxVs. This reduces the load for the operator and provides a more robust control algorithm for the UxVs.

Finally, this algorithm was tested in a laboratory environment to demonstrate the capability, and the results were plotted and shown in this paper.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Dr. Oleg Yakimenko for his continuous support and motivation throughout my thesis process. I also would like to thank Dr. Fotis Papoulias and Dr. Gerard Leng for their advice contributed for this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. BACKGROUND

A. GENERAL

There is a trend of demographic transition occurring in the developing world. The world's urban population is increasing four times as fast as its rural population. By 2025, two-thirds of the Earth's population is projected to live in urban areas, and 90 percent of the growth will be in the developing world [1]. Therefore, the needs of military systems are changing to focus upon urban warfare or urban operations (UO). The shifting of traditional warfare from the field to the urban environment drives significant changes in military strategies, particularly the technological development of military systems.

Intelligence, surveillance, and reconnaissance (ISR) operations are of great importance in urban warfare. In order to mitigate the risk in these complex and dynamic environments, an organized military force will attempt to understand as much about its environment as it can, in order to make well-informed decisions and comply with the rules of engagement by identifying their opposing threats. This is known as situation awareness [2]. This capability can be achieved by deploying unmanned systems with sensing and communication payloads. However, in urban environments, conventional systems may not be able to meet mission objectives due to the complex nature of the environment.

A city is more than just a physical environment. There are also political, economic, social, and psychological constraints in the urban environment [3]. The physical constraints involve both natural and man-made structures.

They pose a major hurdle for a conventional unmanned system to perform its ISR missions. Most of the systems rely on radio-frequency sensors and will suffer from interference with other signals, or with themselves, due to multipath propagation effects [2]. The lines of sight are short in these environments and the GPS is often unable to work most of the time as it cannot see the minimum number of four satellites.

Singapore is a country with a typical urban environment. Based on a study carried out in Singapore by a team of career officers from the armed forces [4] on combined unmanned air and ground vehicles operations in the Singapore urban environment, the environment can be classified into five types of zones as shown in Table 1. This classification is based on tactical considerations for fire support, maneuver, cover/concealment, and command/control.

UTZ	**Descriptions**
Type A	Dense, quasi random construction (e.g., CBD)
Type B	Close, orderly blocks (e.g., pre-WWII shop houses)
Type C	Dispersed residential areas (e.g., landed housing estates)
Type D	High-rise areas (e.g., HDB housing estates)
Type E	Industrial/ transportation areas (e.g., Jurong island)

Table 1. Urban Terrain Zone (UTZ). From [5].

In order to develop situational awareness in an urban environment, multiple unmanned vehicles can be used in a collaborative manner to counter these problems and acquire timely information on the environment for the urban warfare commandants. As highlighted by [6], unmanned vehicles can continuously meet the operator's requirements and needs during or before the battle. It is expected that they will become an indispensable support element for a wide range of battles in the future.

In [6], a comparison was made to evaluate the type of unmanned vehicles that are most suitable for each type of mission. The pros and cons were discussed with regard to use of a quadrotor and fixed-wing unmanned aircrafts. The fixed-wing UAV is limited by its inability to fly low and hover, requiring a take-off and landing distance requirement, as compared to a quadrotor which has a complex design but offers better mobility and lower endurance. Since an urban environment has limitations in its air space, this poses a challenge for fixed-wing UAVs to be able to fly at a safe altitude and spot the target. Thus, quadrotor UAVs are explored in the following section which focuses on the various UAVs being deployed in urban environments.

B. URBAN ENVIRONMENT UAVS

Various types of UAVs catered for the urban environment have been developed over the years. This section will discuss these UAVs and compare their performance.

1. Qube UAS

Figure 1. Qube UAS by Aerovironment. From [7].

The Qube UAS is a rugged and reliable small UAS primarily used for public safety purposes. This system can be easily stored in the trunk of a car and assembled within five minutes. It provides real-time video transmission to the operator and is able to carry out missions such as searching for suspects or missing persons, standoff or hostage situations, accident or crime scenes, fire-fighting support, disasters, and explosives or bomb disposal response [7].

2. SQ-4

Figure 2. SQ-4 by Middlesex University. From [8].

A team of engineers from Middlesex University developed the UK's first lightweight outdoor flying UAV

which can fit in a soldier's backpack. This UAV is a remotely controlled vehicle which provides real-time footage to goggles worn by the operator. It is able to hover quietly and perch on objects while performing persistent surveillance over an area [8].

3. Draganflyer X6

Figure 3. Draganflyer X6 by Dragonfly Innovations Inc. From [9].

The Draganflyer X6 is small enough to fly indoors and has a unique design to maximize thrust which helps reduce the sound output to only 60 decibels. It is transported in a soft shell pack with a military grade backpack. The UAV provides real-time video as well as telemetry to the operator. The system allows multiple interchangeable camera modules which includes a thermal imaging camera [9]. It also applies the same concept as SQ-4 with the use of video goggles and a remote controller to control the UAV.

4. Shrike UAV

Figure 4. Shrike by Aerovironment. From [10].

The Shrike VTOL system is designed for front-line day or night ISR missions. It is able to operate in hover-and-stare or perch-and-stare modes while transmitting real-time information to the common ground control station (GCS) via a digital data link. It weighs about 2.27 kg and is able to hover up to 40 minutes. It also has the ability to perch in discrete locations, from which it can transmit for several hours before returning back to base [10].

5. Ghost UAV

Figure 5. Ghost by IAI. From [11].

Ghost UAV, developed by Israel's IAI Malat, is a rotary mini-UAV system designed for special operations in dense, mountainous, or urban terrain. It is able to carry up to 600 grams of payload and is packed in a suitcase carried by a single soldier. It also has the ability to operate non-line-of-sight missions. The UAV is able to perform automatic take-off and landing, and operates with EO/IR payloads [11].

6. Aeryon Scout

Figure 6. Aeryon Scout by Aeryon Labs Inc. From [12].

Aeryon Scout, developed in Canada by Aeryon Labs Inc is an all-weather UAV equipped with a gyro-stabilized payload. It requires no launch equipment and enables fixed hover for precise observation for covert operations. It provides simple, touchscreen, waypoint-planning controls for the operator and therefore requires minimal training for soldiers [12].

7. RQ-16A T-Hawk

Figure 7. RQ-16A T-Hawk by Honeywell. From [13].

RQ-16A T-Hawk is a ducted-fan VTOL micro-UAV suitable for backpack deployment and single-person operation. This UAV is in operational service in the United States and acts as a good force multiplier for operations in the Iraq war [14]. This UAV is lightweight and is coupled with a ruggedized control station. Real-time video is transmitted to the control station to provide support for ISR missions.

8. Specifications Study

Table 2 was generated from research performed to compare the specifications of various types of urban environment UAVs. There were few similarities among the various UAVs, as discussed. These UAVs usually operate on battery instead of gasoline engines, since they generate less noise. However, this reduces the endurance of the UAVs. This flaw can be overcome by having perch capability where the VTOL UAV is able to land and perform persistent surveillance on an area. The typical speed is around 35–55

kilometers per hour and has an operating altitude ranging from 120 to 3000 meters. As the UAV is generally operated by either a single person (or at most two operators), the weight of the UAV has to be light. The range of weight is from 0.4 kg to 8 kg.

UAV Name	Weight (kg)	PL Wt.(g)	Endurance (min)	Speed (kph)	Range (km)	Op. Alt. (m)
QUBE	2.5	?	40 (with PL)	?	1	152.4
SQ-4	0.21	50	15	18	1.5	120
Dragan-flyer X6	1.5	500	20 (w/o PL)	50	0.5	2438
Shrike	2.5		40	55	5	?
Ghost	4	500	30	35	4	?
Aeryon Scout	1.3	400	25	50	3	500
RQ-16A T-Hawk	7.9	500	50	74	10	3048

Table 2. Specifications of urban environment UAVs

Figure 8 shows the segmentation of the UAVs in terms of their weight and endurance. Urban Environment UAVs generally requires least amount of weight due to the nature of the operation and the limitation of a quadrotor. A typical urban operation has to be swift and therefore does not require long endurance. A quadrotor is usually limited in its weight capacity as it trades off its high maneuverability. In some cases such as search and rescue or surveillance missions in urban environment, longer

endurance may be required. Thus, there is a capability gap of having an UAV with high maneuverability to fly in urban terrain with long endurance.

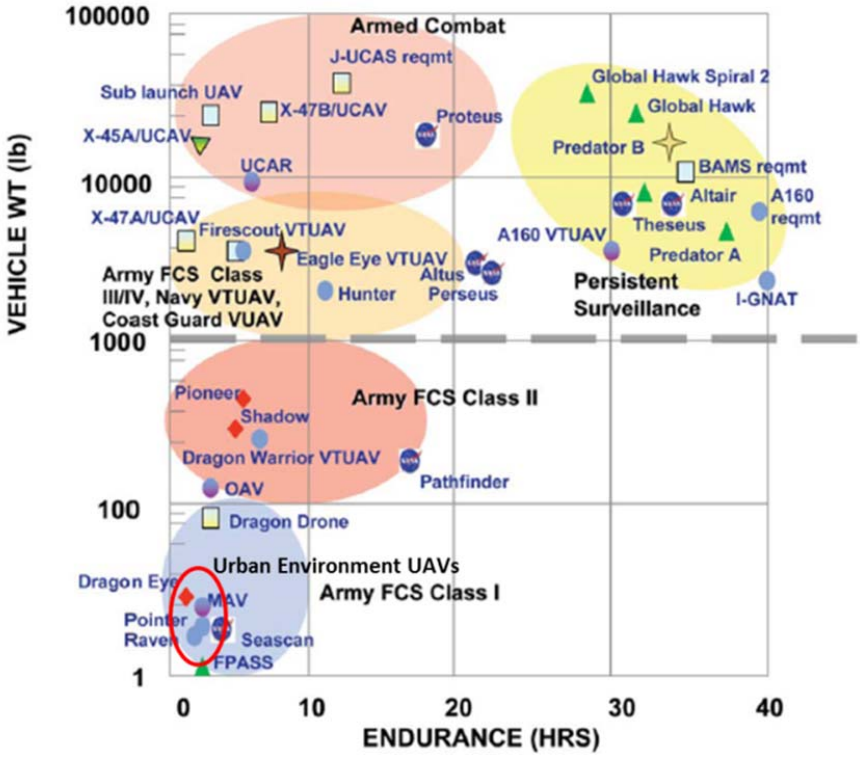


Figure 8. Segmentation of UAVs in terms of endurance. After [2].

Figure 9 shows a plot of the weight vs endurance of the UAVs based on Table 2. It is desirable for the UAV to be on the upper left corner of the plot with low weight and high endurance. The less weight the UAV, the more portable the UAV is and it also indirectly increases the endurance of the UAV. There are 3 main contributors to the weight of the UAV namely frame, sensors and battery.

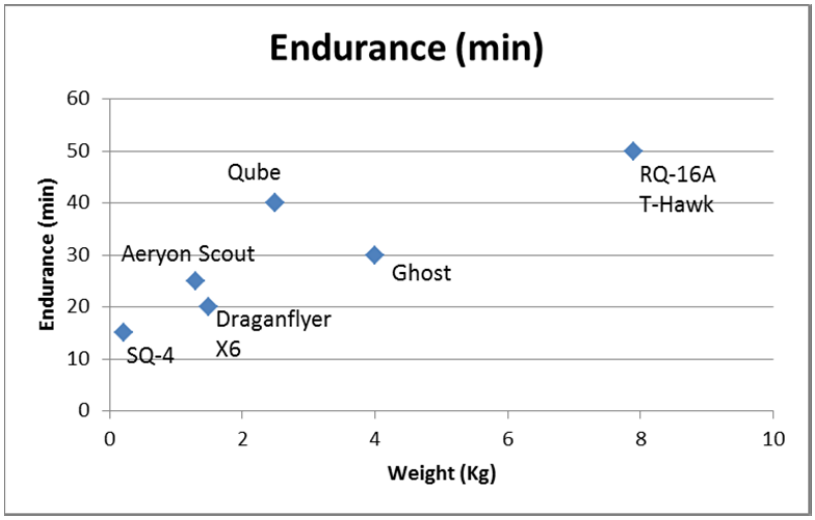


Figure 9. Weight vs endurance of quadrotors

Taking an urban environment UAV, Spreading Wings S800 [15], for analysis as it has available information on the weight distribution of the UAV. The overall mass and weight ratio of the various components of the UAV can be calculated as shown:

$$M_o = M_{frame} + M_{sensor} + M_{battery}$$

$$M_o = 1.1 + 2.5 + 1.5 = 5.1kg$$

$$\varepsilon_{frame} = \frac{1.1}{5.1} = 0.22$$

$$\varepsilon_{sensor} = \frac{2.5}{5.1} = 0.49$$

$$\varepsilon_{battery} = \frac{1.5}{5.1} = 0.29$$

The frame includes the ESC, motor engine and propeller of the UAV. With the comparison of the weight distribution, sensor (or payload) requires the most weight, followed by the battery. Thus, by miniaturizing the sensor and battery of the UAV without affecting the performance of the power will improve the endurance of the UAV. A company named

LaserMotive has developed a wireless power technology solution that is able to extend the battery life of UAV by using lasers. A demonstration was carried out in 2010 with Pelican Quadrotor equipped with a light 5-minute battery and was continuously powered with laser for 12.5 hours [16]. Figure 10 shows a comparison of specific power and energy density for various power sources. Laser power has excellent energy density and power density as compared to the other power sources. The only limitations are line of sight and range with this choice of power.

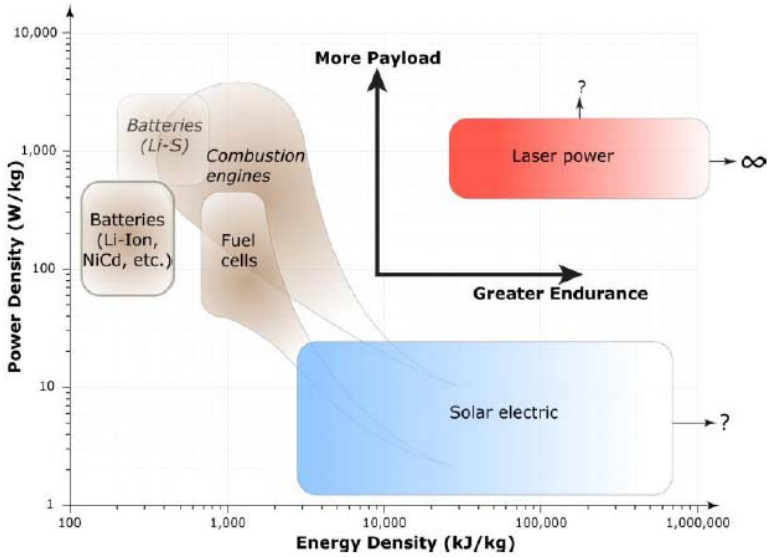


Figure 10. Power and energy density for various power sources. From [17].

Sensors require most of the weight capacity in an urban environment UAV. Tradeoff between the performance of the sensor and weight is often required in the selection of sensor. With higher performance of sensor such as stabilized EO/IR gimbaled camera may require 2.8Kg to 5Kg, whereas a simple digital camera's weight ranges from 100g

to 300g. Thus, the choice of the sensor varies with the application and mission of the UAV.

C. SENSOR INTEGRATION

In order for UAVs to navigate in the complex urban environment, they must be equipped with multiple sensors and robust control algorithms to control the UAVs, and operate in them in a network-centric manner to perform the mission effectively. The U.S Department of Defense defines an unmanned aircraft as an aircraft or balloon that does not carry a human operator and is capable of flight under remote control or autonomous programming. An unmanned aircraft system is also defined as that system whose components include the necessary equipment, network, and personnel to control an unmanned aircraft [13]. By employing multiple UAVs and sharing sensor information, greater coverage can be achieved efficiently and higher success rates can be achieved.

One of the primary functions of a UAV is to collect data and provide information to the user. A UAV system typically includes a Ground Control Station (GCS) which controls and commands the UAV. The GCS can be a mobile station or a fixed station. It collects data from the UAV and translates it into useful information for the operator. In order to transmit data to the GCS, the UAV must carry sensors and payloads to collect the data. A typical UAV (specifically, a quadrotor) sensors suite consists of a 3-axis gyroscope, a 3-axis accelerometer, a 3-axis magnetometer, pressure sensors, a sonar sensor, a GPS unit and a payload. Besides the sensors, the Guidance, Navigation, and Control (GNC) algorithms are required to

provide the necessary autonomy for the UAV and data fusion. There are two main sensor integration sources involved: hardware and software. Besides the physical sensor integration within the UAVs, communication is crucial in passing the sensor information between the UAVs and the control station as well as knowing the positions of friendly UAVs. By employing medium altitude or high altitude UAVs to relay communications, the system can offer line-of-sight (or near line-of-sight) links to control stations via the UAVs, or even links to commercial satellites that are over the horizon from ground-based jammers [3].

1. Sensor Hardware Integration

The hardware has a physical integration which includes sharing of processor hardware, power supplies and aperture integration [14]. Future sensor payloads will be combining various functions of payloads to create a more robust and higher performance sub-system. For instance, SELEX Galileo has developed the PicoSTAR featuring compact design and a fully integrated RF and EO sensor payload. This payload delivers radar, electronic surveillance, and electronic attack and communication functions.

Figure 11. PicoSTAR by SELEX Galileo. From [14].

2. Sensor Software Integration

Sensors software integration, in this context, refers to a sensor data fusion which can be achieved by either gathering sensor data from different sensors or receiving sensor data from multiple UAVs and combining them into improved data. At Ohio State University, research was conducted to perform layered data fusion from multi-UAV sensing. The work involved applying an information-theoretic cost function and cooperative optimization method on multiple mini-UAV sensing. This layered data fusion technique was applied on a single video registration, a video registration with a reference image, and the alignment of two video sequences [18].

Another method suggested by Professors Oleg Yakimenko and Gerard Leng to perform continuous surveillance of a target in an urban environment is to use multiple fixed-wing UAVs with a formation flight control algorithm [5]. In order to track a target, an unobstructed line of sight is required with the UAV. However, there may be cases where the geometry of the visible region or constraints on the sensor motion (e.g., limited azimuth angle of the payload, turn radius of the UAV) results in one UAV being unable to track the target. Therefore, the cooperative deployment of UAVs can be implemented to overcome this problem.

3. UAV Relays

Much research has been carried out over the years search for methods to relay communications over the air. As suggested by [19], a project was carried out to determine the suitable placement of relay UAVs through one or more intermediate relay UAVs passing information to base

stations. Similar research was carried out by [20], which takes into account mission-specific quality measures and the number of UAVs allocated to relay communication. In [21], a method for planning a route for a relay UAV, given the known route of the surveillance UAV, was proposed. This method assures communication at certain time points and suggests a valid relay UAV route as a solution to the problem.

Northrop Grumman developed a system called the Battlefield Airborne Communication Node (BACN), which is installed onto the Global Hawk UAV and provides a persistent gateway in the sky that receives, bridges, and distributes communication among all participants in a battle [22]. It provides real-time information flow between similar and dissimilar tactical data links in both line-of-sight, and beyond line-of-sight, situations.

II. SYSTEMS ENGINEERING CONSIDERATIONS

A. PROBLEM DEFINITION

As discussed in the background section, there is a global trend of increasing urban population as more developing countries are evolving. Terrorists can easily hide themselves in the complex and dynamic urban environment and find opportunities to strike major blows in these areas. This poses a problem for both the military and government in countering this type of attack. Thus, intelligence gathering becomes an important function for urban warfare.

Besides terrorism, natural disasters in cities have also claimed many lives, and as the technology advances more sophisticated systems are being deployed for search and rescue missions. These systems had not only helped to save many lives, but have also brought the importance of technology in this area into focus, along with the risk mitigation benefits that the systems are able to provide.

1. Boundaries

Operating unmanned systems in an urban environment presents challenges related to physical, political, economic, social, and psychological boundaries. The physical boundaries include physical entities in the environment: buildings, roads, highways, ports, rails, airports, subways, and sewage lines [23]. Figure 12 illustrates an urban terrain within the context of urban

warfare. These structures present issues for communications with various systems due to multi-propagation effects and line-of-sight issues.

As for the political boundaries, there are legal norms and restrictive rules of engagement (ROE) that the country has to adhere to in order to satisfy public and diplomatic pressures. According to [3], the international law of war can be reduced to the following key concepts: military necessity, humanity, proportionality, and distinction (or discrimination). These are factors that military personnel have to consider while the enemies (terrorists for example) do not need to concern themselves with these factors. They can disguise themselves as civilians in the cities.

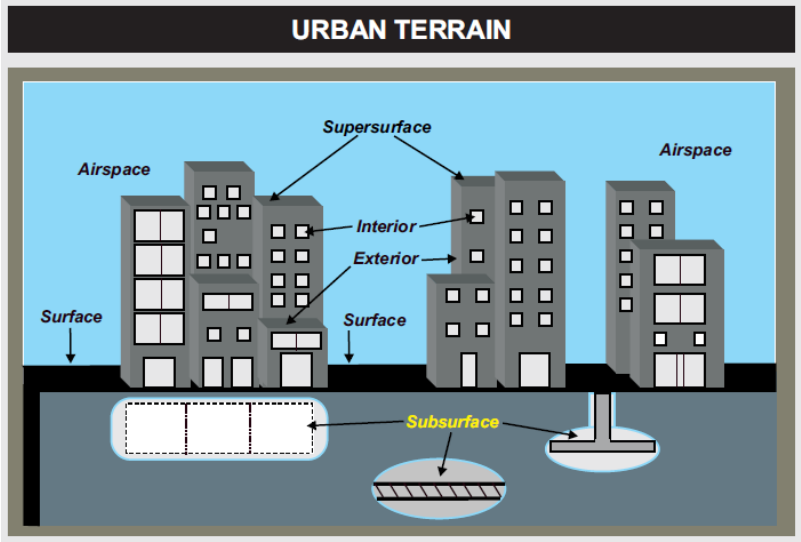


Figure 12. Urban terrain in military context. From [23].

With regard to economic factors, a large amount of resources is needed to develop a system capable of operating in an urban terrain. This capability will be

required in countries with large urban populations that also possess the economic power to develop sufficient capability to operate these systems in urban areas. The prospects of economic prosperity may fall significantly in the event of conflicts; and it is the duty of civilians not the military to restore this prosperity. However, the military is responsible for creating security conditions that make growth and development possible [23].

From a social aspect, language barriers may be an issue while providing foreign humanitarian aid during disasters or war. The system interfaces may communicate with unknown languages or with friendly forces that require translators, which can cause delays.

Due to the high population density in urban areas, the fear experienced by civilians can be as deadly as a stampede or the blockage of evacuation channels that could take place in the event of a crisis. The trust of the people or general public needed to support the operation of unmanned systems remains highly doubtful. It will take time for the technology to prove itself; as the technology matures, people will start to appreciate, trust, and embrace it.

Within these boundaries, the human factor is a key aspect to be considered. Human system integration should be considered prior to the design of these systems. This includes personnel safety while operating the systems and mitigation measures required if the unmanned systems fail (to avoid, endangering people in the area). Training the users for proficiency in the system is crucial so as to

prevent any misuse of the system which may cause hazards or the possibility of failing the mission.

Logistics is another key area to focus on. The availability of the unmanned systems contributes largely to the success of the mission. This factor can be measured by how fast the unmanned system can be deployed upon mission activation, the downtime of the system, and the turnaround time of the system.

The context diagram shown in Figure 13 illustrates a summary of the boundaries.

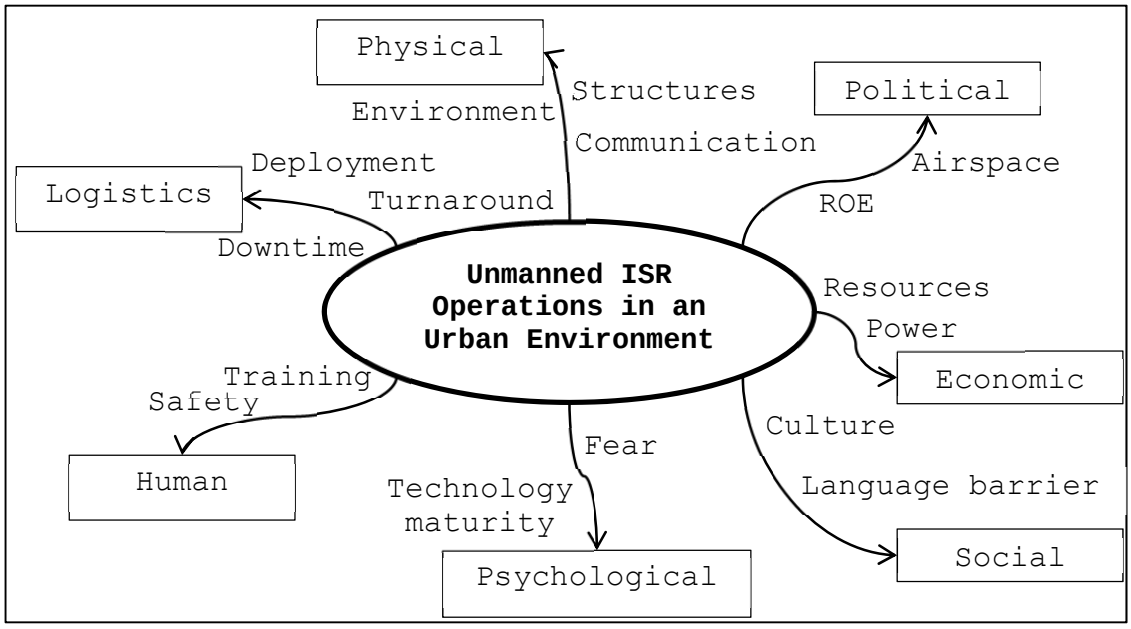


Figure 13. Context diagram for ISR operations in an urban environment

2. Limitations

The most challenging issue operating an unmanned system in a very dense urban environment (such as UTZ Type A and B) is having very limited LOS with the satellites and

control station. This results in difficulty with control and navigation of the unmanned system due to poor or no signals from the satellites, as well as intermittent feedback from the unmanned systems to the control station. There are multiple radio wave transmitting all around the buildings. Besides the multiple radio waves sources, the walls of the buildings can cause the signals of the various unmanned systems operating to weaken due to multipath propagation effects.

There is an unlimited set of obstacles to be considered in an urban environment: they are always changing, dynamic, and can be of any form. The unmanned system has to be robust enough to adapt to the environment. Data about the environment that is both sufficient and up-to-date is required prior to the operation. The flight path of UAVs in the environment is highly likely to be along the flight paths used by commercial planes. Therefore, in order for the UAVs to carry out their mission, they have to operate within a certain altitude range.

Technology limitations can also be a factor in developing suitable unmanned systems for the urban environment. There may be requirements and system engineering analysis performed to develop a system which is able to fill the capability gap for the urban environment surveillance mission, but if the technology has not yet advanced sufficiently, the system cannot be developed.

3. Constraints

One of the greatest concerns is clearing the airspace to fly UAVs in an urban environment. This is due to the safety concerns of deploying UAVs in a crowded environment

with the possibility that the vehicle may drop from the sky and endanger people in the area. There may be a need to evacuate personnel from the area for a mission. This would involve the public as well as the commercial companies in the area. Strong resistance to evacuation may occur. Thus, besides proper process and procedures for the evacuation being ready and in place, the duration of the operation has to be as swift as possible.

The rules of engagement and government regulations play a part in setting the constraints of the system. Depending on the type of mission or situation, the mission commander has to react accordingly, since civilian safety is of top priority. Secondary considerations are the minimizing of collateral and environmental damage. Thus, the design of the system has to take into account these rules and regulations.

Based on a table created by [23], engaging or operating in an urban environment has the greatest challenges based upon the following: number of civilians, infrastructure, environment, rules of engagement, ranges, avenues of approach, freedom of movement, communication restrictions and logistic requirements. This assessment compares the urban aspects with those in the desert, jungle, and mountain environments as shown in Table 3.

Aspect	Urban	Desert	Jungle	Mountain
Number of civilians	High	Low	Low	Low
Amount of valuable infrastructure	High	Low	Low	Low
Multidimensional operational environment	Yes	No	Some	Yes
Restrictive rules of engagement	Yes	Some	Some	Some
Detection, observation, engagement ranges	Short	Long	Short	Medium
Avenues of approach	Many	Many	Few	Few
Freedom of vehicular movement and maneuver	Low	High	Low	Medium
Communication functionality	Degraded	Full Capable	Degraded	Degraded
Logistics requirements	High	High	High	Medium

Table 3. Comparison of operations in urban and other environments. From [23].

4. Scope

The aim of this thesis is to develop a solution for an urban environment ISR unmanned system. The scope is to cover an ISR mission, as these types of missions are considered one of the keys to overall mission success and are expected to continue to grow in the near future with the use of unmanned systems. The urban environment setting

defines the boundaries that the system must look at in the solution space and develop a network-centric solution to provide the capability of operating ISR missions within an urban terrain. This environment poses many challenges, as discussed in the earlier sections, such as: limited ranges, a dynamic environment, and a large number of civilians, as well as logistics requirements and communication restrictions. Certain risks are also identified, such as human safety and the acceptance of unmanned systems by the general public. These risks must be taken in account in the near future as unmanned systems prove their maturity and capability.

B. STAKEHOLDERS AND NEEDS ANALYSIS

1. Stakeholders Identification

A number of stakeholders can be identified that are involved in the system. They can be classified as one of two types: key stakeholders or general stakeholders. The key stakeholders of the system are primarily the users, designers, and policy makers. These key stakeholders are identified as: the government, who will set the policies, rules, and regulations of operating unmanned systems (and also the standards for the urban environment, such as the height limits of buildings, road width, etc.); users/operators can include either the police or military, depending on the nature of the mission (whether the mission is one of public safety or urban warfare). General stakeholders are any others who may be indirectly involved in the operation of the system. General stakeholders include the civilians or general public, as well as commercial companies.

2. Needs Analysis

Well-informed decisions are of high importance in urban warfare, which brings ISR capability into focus. There are four main categories of urban operation missions: Law enforcement, emergency measures, fire, and tactical surveillance. These missions will be discussed in more detail in a later section. There is also the desire to have higher autonomy, so as to reduce the number of operators and minimize human interventions. The various operational capabilities based on the perspectives of the stakeholders for ISR are identified as follows [23]:

- Visualize the operational environment
- Provide situational awareness
- Process intelligence and disseminate it to operating forces in real time
- Provide timely intelligence and information to support decision making

From these identified operational capabilities identified, an overarching capability need statement is derived as follows: "complex urban environment with limited protection capabilities need timely intelligence to counter asymmetric threats."

C. CONCEPT OF OPERATIONS

To summarize the previous discussion, the system should be able to provide a solution to meet the specified capability requirements. Figure 14 illustrates the system solution for providing the capability for ISR missions in an urban terrain. A central control station takes up the

role of a central coordinating and decision-making process. This central control station will only need to communicate with the relay UAV to the respective vehicle ground stations (which includes the UGV and the mission UAVs or Quadrotors). The central control station will be remote from the vehicle ground stations and is usually not within line of sight. The operators of the vehicle ground stations will provide the mission area, most recent map data, flight altitude, arrival schedule, type, and number of UAVs to control from the ground station and control these UAVs within close proximity around the mission area. From this information, the ground station performs resources allocation and calculates the locations of the waypoints based on the UAV's camera field of view and altitude. It then sends the waypoints to the UAVs to scan the area. The UAVs are equipped with a real-time trajectory generation based on the direct method of calculus of variations [24] which are capable of performing dynamic retargeting and obstacle avoidance as needed. Due to the dynamic environment, the map data may not always be reliable, and this control method allows the UAV to perform avoidance of obstacle or friendly UAVs that it spots in its camera. The relay UAV is used to relay communication from the mission UAVs and UGVs to the vehicle ground station and central control station as well as providing GPS information to the mission UAVs [25]. Since UAVs move much faster, they are deployed to scan the area first. Once the target/IED is identified, a UGV can be deployed to neutralize it.

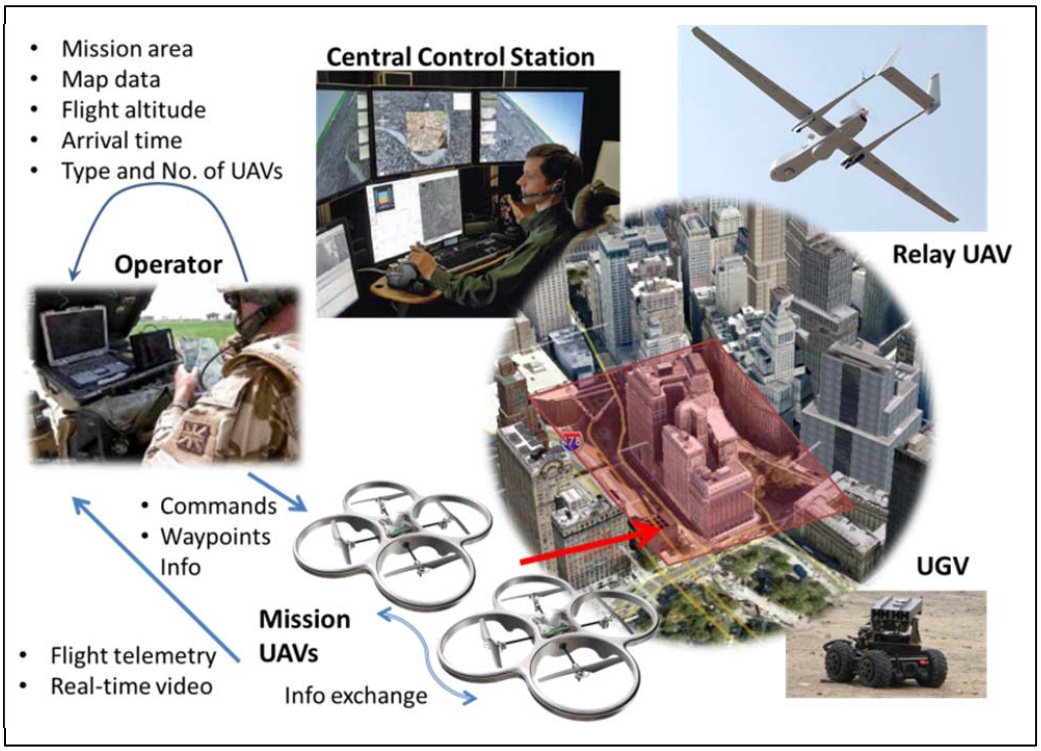


Figure 14. Concept of operations

D. FUNCTIONAL ANALYSIS

Based on the operational capabilities identified by the stakeholders and the operational concepts developed, a list of key functions is derived as follows:

- Provide centralized mission command and control
- Integrate sensor data to produce sensible information
- Detect and identify a target
- Avoid obstacles and friendly UAVs
- Real-time telemetry and video
- Assure real-time telemetry and video streaming

- Provide communication between the central control station and vehicle ground station
- Provide communication between relay UAVs and the central control station and mission UAVs
- Provide a means to neutralize target
- Provide command and control over mission UAVs
- Maneuver around urban terrain

From the list of functions derived, three main functions can be identified as: "to communicate," "to detect and identify target" and "to maneuver around an urban terrain." These functions are decomposed into a hierarchy chart as shown in Figure 15.

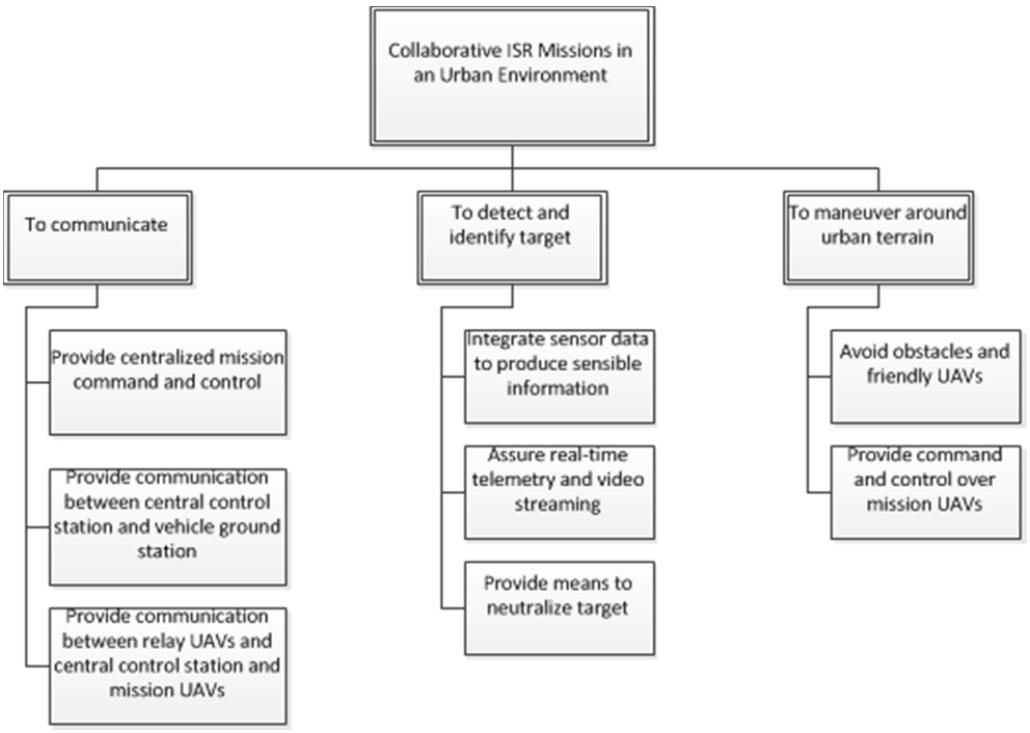


Figure 15. Functional decomposition

E. INTEROPERABILITY CHALLENGES

Interoperability is defined as the ability to synergistically operate in the execution of assigned tasks and exchange information and services directly between systems and users [13]. In systems that operate collaborative ISR missions in an urban environment, the system is comprised of a 'system of systems.' The integration of one system with another system brings about challenges such as joint operations, joint interoperability, and the dimensions of distributed command and controls [26]. Based on the concept of operations, there are at least five systems requiring interoperable with one another. They can be identified as the central control station, the relay UAV, the mission UAVs, the UGVs and the operator control stations. The interoperability of these systems and processes requires three key factors: connectivity, coupling, and cohesion [26].

Connectivity is defined as the interaction, or the facilitation of interaction between objects or processes [26]. These systems can be intermediate nodes; as long as they have inter-connectivity, they possess the pre-requisites for interoperability. The systems must be connected with common data link communication. In order to overcome the LOS issues with the operation team (mission UAVs/UGVs and operator control stations), they will be communicating with the relay UAV to the common control station.

Coupling is defined as the degree of dependency between objects or between processes, and cohesion is defined as the degree to which the objects or processes

relate to each other [26]. These two factors are important as they "perform" the interoperability of the system. Coupling refers to the systems being able to acquire the correct EMMI (Energy, Matter, Material and Information) [26] in a timely and meaningful fashion. Cohesion refers to the ability of the system to carry out the intended design and action desired. For instance, the commander of the operation can be at the central command station directing orders to the operators at the urban terrain using the UAV relays. Connectivity has to first be available in order for these commands to be transmitted to the operators. Coupling comes to play when these orders arrive to the operator accurately and within the specified time interval. Cohesion will depend on whether the operators carry out the mission which the commanders ordered them to perform. This same logic applies to the operators and the mission vehicles.

Other interoperability challenges that must be considered include the mission requirements, standards such as the communication protocols in urban environment, and operating in the national air space.

F. RISKS

Risk as defined in the aviation is the likelihood of a hazard causing an undesirable incident combined with the severity of the incident [27]. The most severe incident involves either death of injury to persons. With the focus of this type of hazards, three major aggregate risk categories are listed as [27]:

a. Death or injury of persons **on board subject aircraft,** resulting from a mishap,

b. Death or injury of persons **on board another aircraft,** resulting from a mid-air or surface collision between two or more aircraft/ground vehicles,

c. Death or injury of persons **on the ground (**not in an aircraft or vehicle involved with a collision) resulting from a mishap or collision.

In this context, the focus is on unmanned aircraft. Thus, the first risk is eliminated as there will be no one on board of the aircraft.

Considering the second and third risks of mid-air collision and collision of person on the ground, a global risk matrix is used to assess the risks involved as shown in Figure 16.

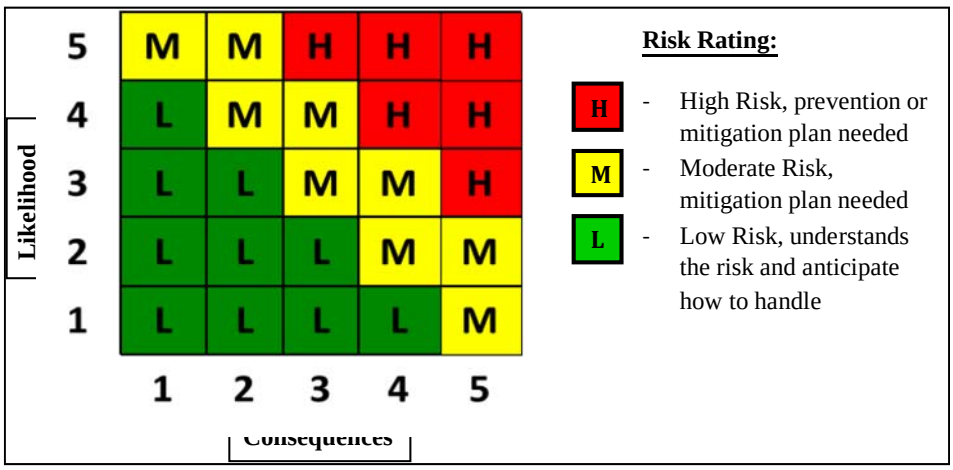


Figure 16. Global risk matrix

The definitions for the likelihood and consequences are tabulated as shown in Tables 4 and 5.

Level / Likelihood	Description
5 / Nearly Certain	Risk events are imminent and cannot be avoided under current conditions – incapable process
4 / Highly Likely	Expects risk events and most of them are likely to occur – incapable process
3 / Likely	Anticipates risk events but may not avoid them – marginally capable process
2 / Unlikely	Usually avoided or resolved risk events in similar cases – capable process
1 / Remote	Effectively avoid or resolve risk events using standard practices – highly capable process

Table 4. Likelihood description

Level / Consequences	Description
5 / Catastrophic	Risk events that results in death of persons
4 / Serious Injury	Causes serious injuries to the persons
3 / Minor Injury	Causes minor injuries to the persons
2 / Create commotion	Events that develop fears and results in commotion to the persons
1 / Safe	Has no consequences and is safe

Table 5. Consequences description

The remaining two risks that could occur are summarized into Table 6 with the addition of mitigated courses of action, root cause and the severity of the risk (L – Likelihood, C – Consequence) based on the matrix.

Description	Root Cause	L	C	Risk Rating	Mitigation	L	C	Risk Rating
Mid-air collision with other aircraft / obstacle	Loss of controls, Loss of visual sight of aircrafts	2	5	**Medium**	Collision avoidance algorithms, See and Avoid capability, Smaller size UAVs	1	4	**Low**
Collision to people on ground	Loss of controls	2	5	**Medium**	Design safe flight termination capability, Safe design on UAV that poises minimal risk even when struck by it	2	3	**Low**

Table 6. Risk matrix for risks (b) and (c)

This risk of mid-air collisions could occur when the UAV either loses its flight control or the operator loses sight of the UAV or other aircraft. This could result in death of person if the UAV collide with another manned aircraft. This thesis focuses on tackling this problem by developing a control algorithm that is able to avoid other

aircrafts and obstacles and therefore reducing the likelihood of it occurring. In order to reduce the consequences, smaller UAVs can be used. From [27], an aircraft is designed to withstand a bird strike. So if the UAV is able to design to be small enough that the consequences would be no worse than a bird strike, the risk consequence will be reduced significantly. Thus, the risk rating with the mitigation will be dropped to "Low" from the initial assessment of "Medium" risk.

The risk of the UAVs crashing onto the ground and colliding on the people has a risk rating of "Medium." This is due to the high consequence rating as the result can be catastrophic. This consequence can be mitigated by establishing design standards of UAVs to have a flight termination capability that will reduce the risk of injury to people on the ground. For instance, an airbag/parachute can be deployed to land the UAV, in the event of loss of flight control. Another method can have design standards to build the UAV such that it poises minimal risk to people on the ground even if they are directly struck by the aircraft. In [27], this scenario is compared with non-aircraft objects such as baseball or golf ball that could prove to be lethal if they struck on a human, but yet they are acceptable in the society. A comparison was made between the lethality between various objects with kinetic energy as shown in Table 7 and Figure 17.

	Baseball	Golfball	Very small UAS (WASP)	Small UAS (DraganFlyer X4)	Park Flyer RC Model Aircraft	Small UAS	Largest Model RC
Weight (lbs)	0.31	0.1	0.95	1.5	2	20	55
Comparison Velocity (mph)	95	170	40	35	60	25	200
Kinetic Energy (J)	128	131	69	81	326	567	99714

Table 7. Kinetic energy of various objects. From [27].

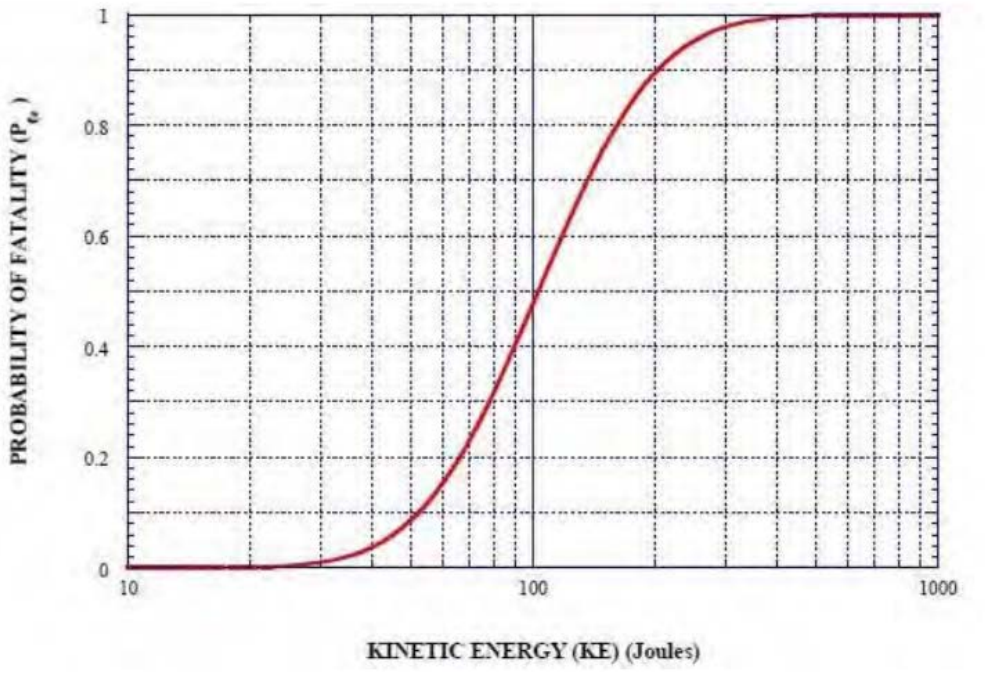


Figure 17. Kinetic energy vs. probability of fatality. From [28].

As shown, both the baseball and golf ball could produce kinetic energy that would be over 50% of the time, whereas small UAVs such as WASP or DraganFlyer X4 could

result only 10% of the time that could cause lethality if they hit an unprotected human at full speed. Therefore, with reduction in the size of the UAVs and better designs with safety considerations in mind, these measures can mitigate the risks of collisions with people on the ground.

G. PROBLEM FORMULATION

One of the key problems in the functions specified is how to provide UAVs with the ability to avoid obstacles and friendly UAVs. In [24], a direct method of calculus of variations was suggested by exploiting the inverse dynamics of a vehicle in the virtual domain (IDVD). In this thesis, the method will be applied on a Quadrotor UAV from Quanser named QBall-X4 in an indoor environment.

1. Types of Urban Operations

From [29], the four main categories of urban operations can be extracted. Figure 18 segregate these missions and break them down further into sub-missions from these four categories.

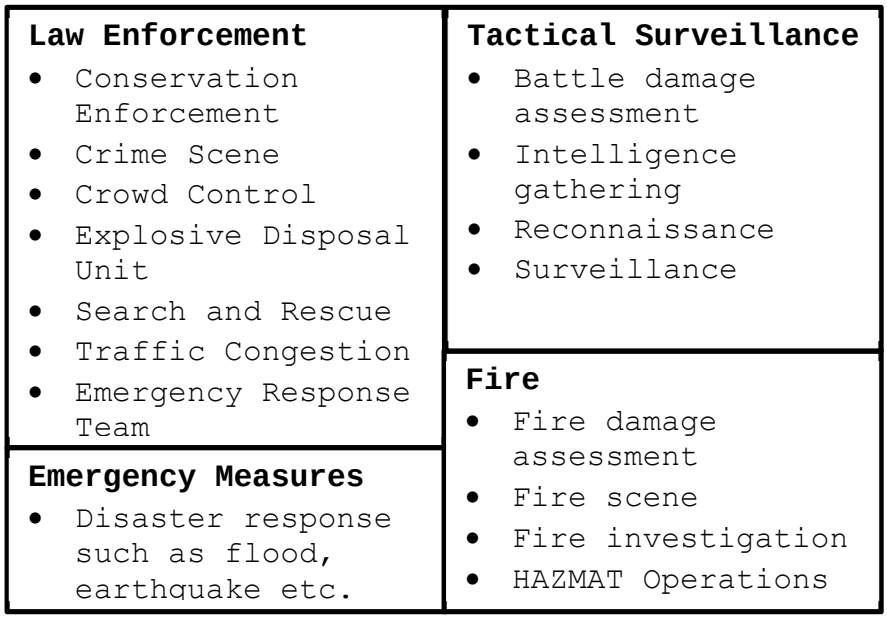


Figure 18. The four main categories of urban operations. After [29].

2. Thesis Design Scenario

There is a suspected Improvised Explosive Device (IED) planted within a vicinity of an urban area in Singapore (as shown in Figure 19). There are three buildings situated around this area. Two UAVs (equipped with an IED detection sensor) will be deployed to scan the area for the location of the IED and transmit the location to a UGV which is capable of disarming an IED. Prior mapping of the area has already been performed and the terrain of the area is known to the system.

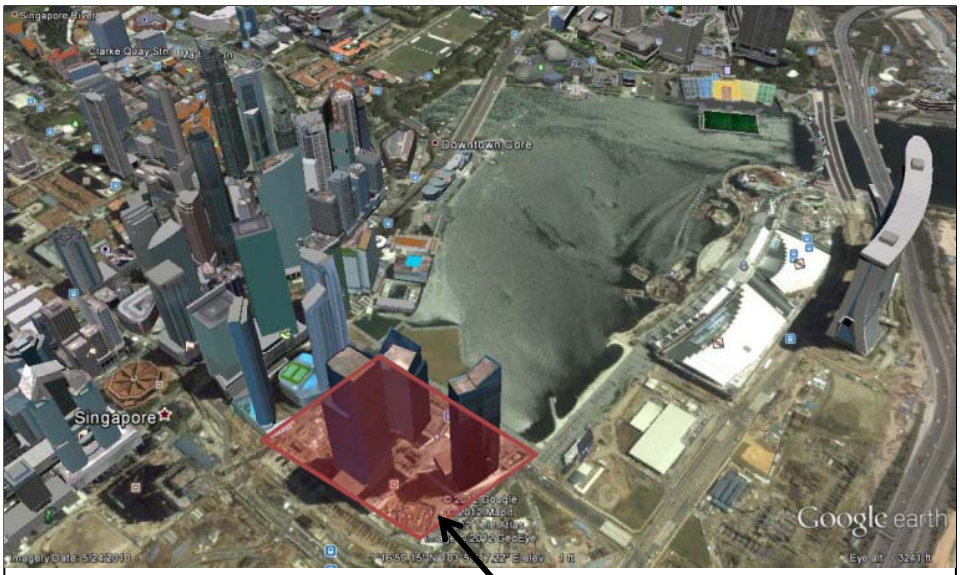


Figure 19. Scenario location

A topographic view of the mission area was taken from Google Earth and this map data is scaled down by 80 times to the lab environment as shown in Figure 20. In order to optimally search for the IED, UAV A will begin its search from one end of the area and UAV B will start its search from the other end of the area. Waypoints were generated automatically by the GCS based on the camera field of view of the UAVs. Optimal trajectory generation in real-time is performed to avoid the obstacles as well as the UAVs.

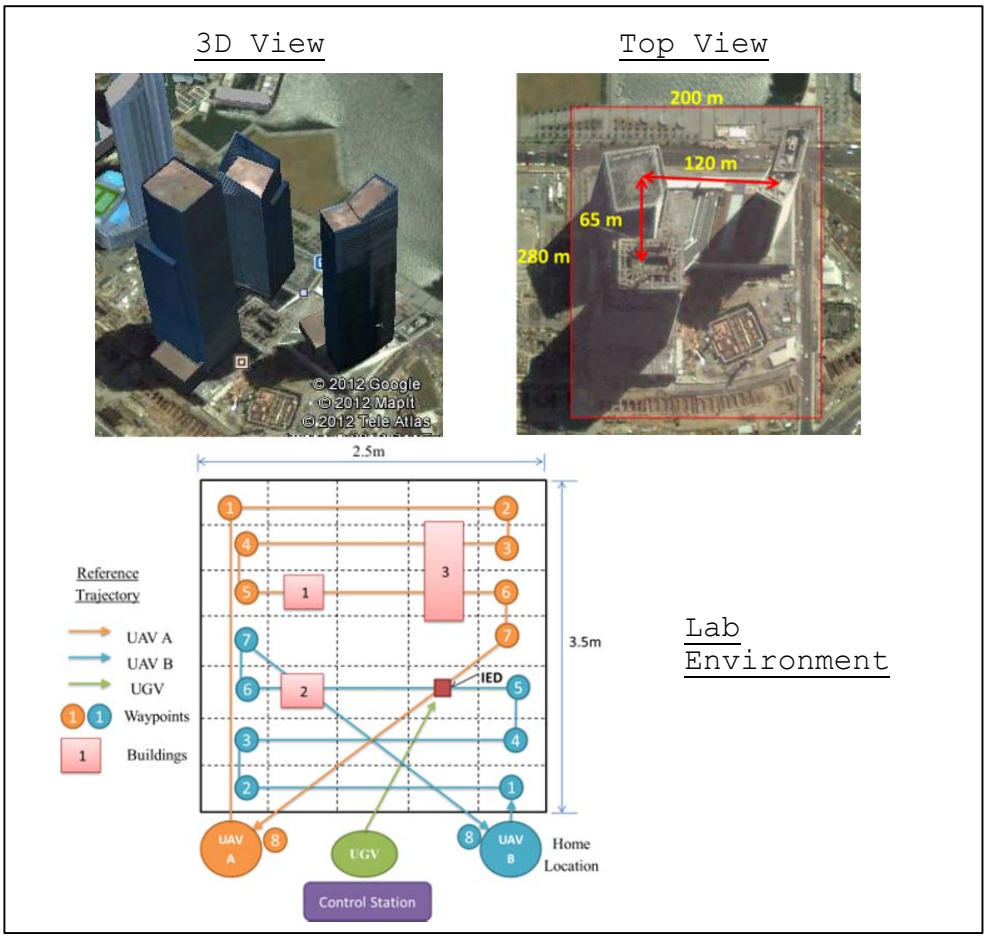


Figure 20. Mapping from actual map environment to lab environment

THIS PAGE INTENTIONALLY LEFT BLANK

III. MODELING

A. QUADROTOR DYNAMICS

The dynamic and kinematic modeling of a quadrotor is presented in this section. Figure 21 shows a quadrotor developed by Quanser Inc. There are a total of four Qball UAVs at the Naval Postgraduate School and these UAVs are provided with a good experimental test-bed with a protective cage to prevent damage. The UAV is equipped with basic autopilot and manual control software, communication and interfaces, which makes the platform a good tool for concept demonstration.

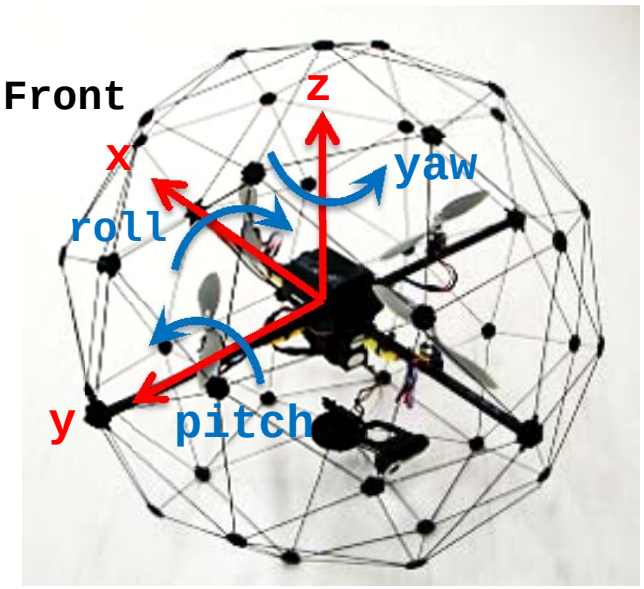


Figure 21. Quanser Qball X-4 UAV. After [30].

1. Coordinate Frames

There are three types of coordinate frames used in this paper: body-fixed frame, Optitrack coordinate frame,

and Earth-Fixed inertial frame U. Figure 21 shows the body-fixed coordinate frame of the UAV where the frame is attached to the center of mass of the quadrotor and rotates with the vehicle. The x axis of the frame is along the axis of the two opposing propellers and pointing towards the front of the vehicle. The y axis points to the left side of the vehicle, and z axis points upwards. The right-hand rule is used to determine the direction of the euler angles of the vehicle. A positive roll direction is counterclockwise about the x axis when facing the quadrotor. This rule dictates the same for pitch and yaw direction. A sticker is pasted on the front of the vehicle frame to indicate the x axis of the vehicle.

The coordinate frame used by the Optitrack system (which serves as positioning tracking system for the UAVs) is fixed on the ground at the center of the indoor lab. More details of the system will be discussed in a later section. Figure 22 shows the coordinate frame in the lab.

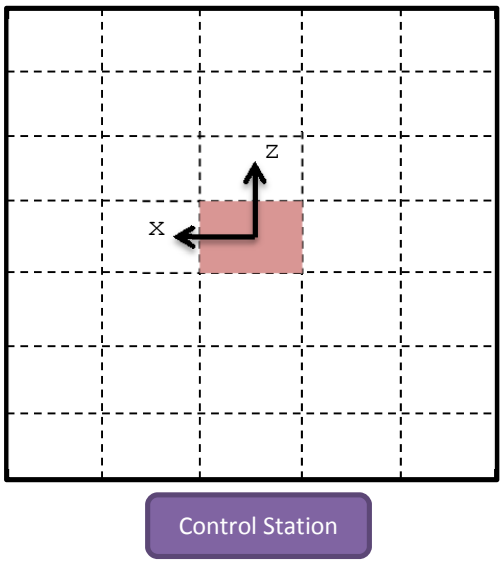


Figure 22. Optitrack system coordinate frame

The x axis points towards the left of the lab from the control station and z axis points away from the control station. Y axis is pointing upwards from the ground. Note that within the UAV Simulink models, the direction of x and z axes are inverted to match the commands given to the UAVs.

In the derivation of the dynamics modeling, the earth-fixed inertial frame U is used. The coordinate frame is of North-East-Down (NED), with the origin at an arbitrary ground point, and it is chosen to be the quadrotor take-off point.

2. Assumptions

Several justifiable assumptions are made to simplify the modeling of the complex dynamics of the quadrotor:

- The Earth is flat and not rotating.
- Constant acceleration of 9.81 m/s^2 due to gravity.
- The quadrotor is a rigid body that does not flex.
- Drag forces are ignored. (Since the speed of the experiment is low, drag forces are negligible).
- Pitch and roll angles of the Quadrotor throughout the flight are small.

3. Model

The quadrotor is controlled by independently varying the speed of four rotors. By changing the torque and thrust of the rotors, different thrust, roll, pitch, and yaw moments are generated to control the UAV. Figure 23 shows

the schematic of a quadrotor and the numbering of the rotors, as well as the directions of the torque and thrust of each.

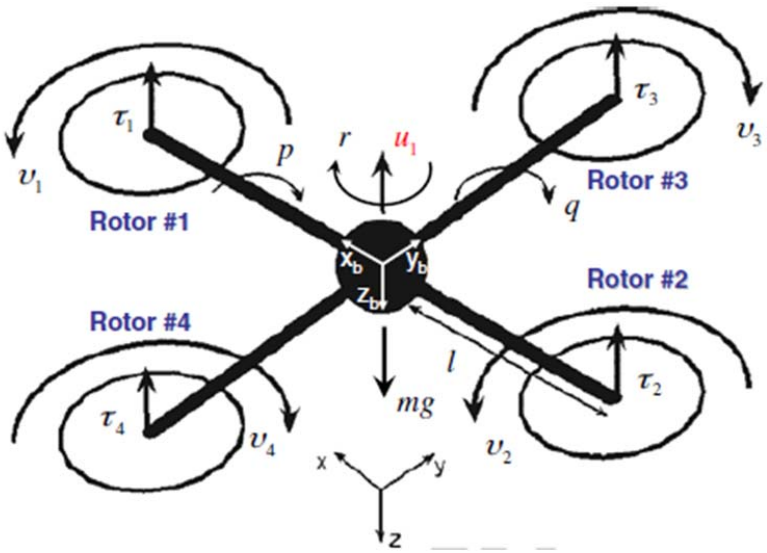


Figure 23. Quadrotor schematic. From [24].

Let $\tilde{u}_i$, v_i, and τ_i be the four controls in the body frame, normalized torque, and normalized thrust for the ith rotor, respectively, where i = 1,…,4. Based on [24], a total normalized thrust in the body frame is given by:

$$\tilde{u}_1 = (\tau_1 + \tau_2 + \tau_3 + \tau_4); \quad (1)$$

A roll moment can be achieved by varying the left and right rotor speeds:

$$\tilde{u}_2 = l(\tau_4 - \tau_3); \quad (2)$$

A pitch moment can be generated by varying the front and back rotor speeds:

$$\tilde{u}_3 = l\left(\tau_1 - \tau_2\right); \tag{3}$$

The yaw moment can be obtained from the difference in the counterclockwise and clockwise normalized torques of each rotor:

$$\tilde{u}_4 = \left(v_3 + v_4 - v_1 - v_2\right) \tag{4}$$

By introducing a twelve-state vector of

$$x = \left[x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi} \right]^T \tag{5}$$

where $\left[x, y, z\right]^T$ is the translational position of the quadrotor center of gravity in the NED frame and $\left[\phi, \theta, \psi\right]^T$ is the attitude vector comprised of roll, pitch, and yaw angle respectively between $\left[x, y, z\right]$ and the body frame. The desired outputs of the system are the translational position and the yaw angle. Thus, by defining the control vector **u** from the total normalized thrust and second derivatives of the Euler angles, and developing the equations of motion by using the rotational matrix, the complete set of equations for the state vector is derived as follows [24]:

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ -\cos\phi\sin\theta u_1 \\ \sin\phi u_1 \\ g - \cos\phi\cos\theta u_1 \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{6}$$

The three controls in the body frame can be derived by applying the rotational matrix from the relationship between the body rates and the Euler rates. Differentiating this relationship (and with an assumption of small rates) produce the controls in the body frame as follows:

$$\begin{bmatrix} \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \end{bmatrix} \approx \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi\cos\phi & 0 \\ -\sin\psi & \cos\psi\cos\phi & 0 \\ 0 & -\sin\phi & 1 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} + \begin{bmatrix} -\sin\psi\dot{\psi} & \cos\phi\cos\psi\dot{\psi} & 0 \\ -\cos\psi\dot{\psi} & -\cos\psi\sin\phi\dot{\phi} & 0 \\ 0 & -\cos\phi\dot{\phi} & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{7}$$

B. SENSOR DATA PROCESSING

Figure 24 shows the system overview of the Quanser Unmanned Systems Laboratory setup [31].

1. QuaRC Real-time control and multi-agent mission development frameworks
2. Ground control station
3. Unmanned aerial vehicles (UAVs)
4. Embedded computer systems and inertial measurement

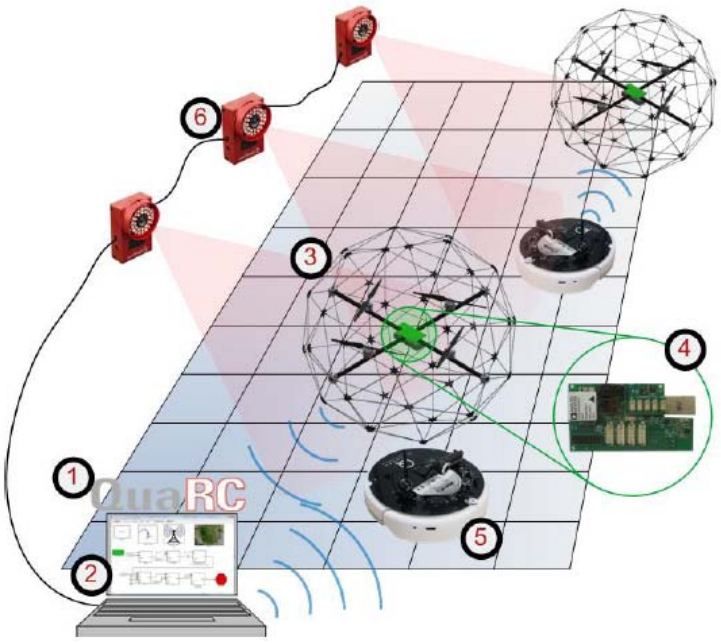


Figure 24. System overview of Quanser unmanned systems laboratory setup. From [31].

The ground control station is equipped with Matlab/Simulink and Optitrack Tool software. Matlab/Simulink software is used for developing the algorithms, control, and communication for the unmanned vehicles. The Optitrack tool is used to perform calibration of the cameras for the localization system. The UAVs and UGVs are equipped with a HiQ data acquisition card and Gumstix processor to perform communications with the ground control station and perform the autopilot function.

1. HiQ Data Acquisition Card / Gumstix Processor

These two important components of the system comprise the "brain" of the unmanned vehicles. They provide the states of the vehicle and telemetry back to the ground control station and to Gumstix. The Gumstix performs the autopilot functions from the codes downloaded from the host

computer and processes the inputs from the sensors. The input/output of the HiQ data acquisition card consists of the following [31]:

- 10 PWM outputs (servo motor outputs)
- 3-axis gyroscope, range configurable for ±75°/s, ±150°/s, or ±300°/s, resolution 0.01832°/s/LSB at a range setting of ±75°/s
- 3-axis accelerometer, resolution 2.522 mg/LSB
- 10 analog inputs, 12-bit, +3.3V
- 3-axis magnetometer, 0.76923 mGa/LSB
- 4 Maxbotix sonar inputs, 1 inch resolution
- Serial GPS input
- 8 channel RF receiver input
- USB input for on-board camera (up to 9fps)
- 2 pressure sensors, absolute and relative pressure
- Input power 10-20V

2. Sensors

There are several sensors installed in QBall UAV. The installed magnetometer has an accuracy of 0.5 mGa/LSB, but was determined to be unreliable due to the magnetic field generated from the electrical wires within the lab [30]. Therefore, the main sensors to control the attitude of the

UAV are a 3-axis gyroscope and accelerometer. The accelerometer has a resolution of 3.33 mg/LSB and the gyroscope is reconfigurable for +/-75°/s, ±150°/s, or ±300°/s with a resolution of 0.125°/s/LSB at a setting of ±75°/s [32].

Due to the compact indoor lab environment, there is a need to fly with precise height control. The pressure sensors are not able to produce such accuracy and therefore, sonar sensor is used to control the height. The sonar used for the UAV is Maxbotix XL-Maxsonar EZ3, which is capable of measuring altitudes between 20 cm and 765 cm with 1 cm resolution [32]. Since the sensor is located at the bottom of the protective cage (shown in Figure 25), correction of the readings is required to offset it to the center of mass of the UAV. This is done by correcting it with the height difference between the sonar sensor location and center of gravity with the pitch and roll readings of the gyroscope and accelerometer.

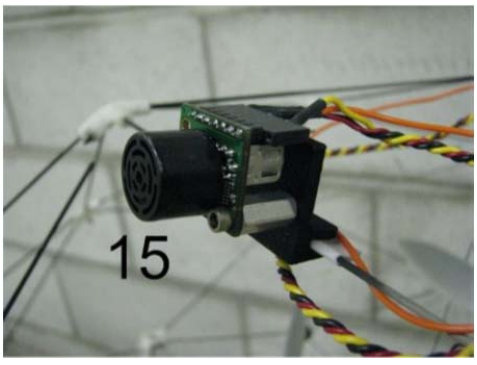


Figure 25. Sonar sensor

Due to the testing of the UAV in an indoor environment, GPS is unavailable. Therefore, the setup

includes a localization system with infrared cameras from Optitrack to provide precise locations of the UAVs and UGVs. The position data is transmitted through an USB connection to the ground control station which will then relay the data over an ad-hoc wireless connection to the UAVs/UGVs. The Optitrack camera system captures an infrared signature from multiple light emitting diodes (LEDs) or reflectors fixed on the UAVs as shown in Figure 26.

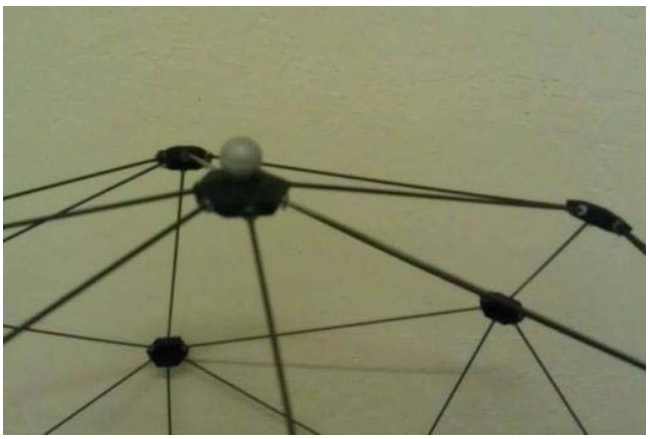


Figure 26. Reflectors on the UAV

The Optitrack system developed by Natural Point makes use of infrared cameras to track the positions of the UAV on the attached LEDs. The system consists of 11 cameras mounted around the lab. The Optitrack vision system has the following features [31]:

- Up to 16 cameras can be connected and configured for single or multiple capture volumes
- Capture volumes up to 400 square feet
- Single point tracking for up to 80 markers, or 10 rigid-body objects

- Typical calibration time is under 5 minutes
- Position accuracy in the order of mm under typical conditions
- USB 2.0 connectivity to ground station PC
- Up to 100 fps tracking

Figure 27 shows the model of the infrared camera used in the lab. Each camera has a field of view of 46 degrees and a resolution of 640x480 pixels at a frame rate of 100 frames-per-second. The cameras were mounted approximately 10 feet from the ground to provide the maximum capture volume as shown in Figure 28. This maximum capture volume will depict the flight boundary for the UAV to fly within the lab environment.

Figure 27. V100:R2 infrared camera

The Optitrack system comes with software called "Optitrack Tool" which provides the user interface for system camera calibration. A calibration procedure created specifically for use in the lab at Naval Postgraduate School can be found in the Appendix.

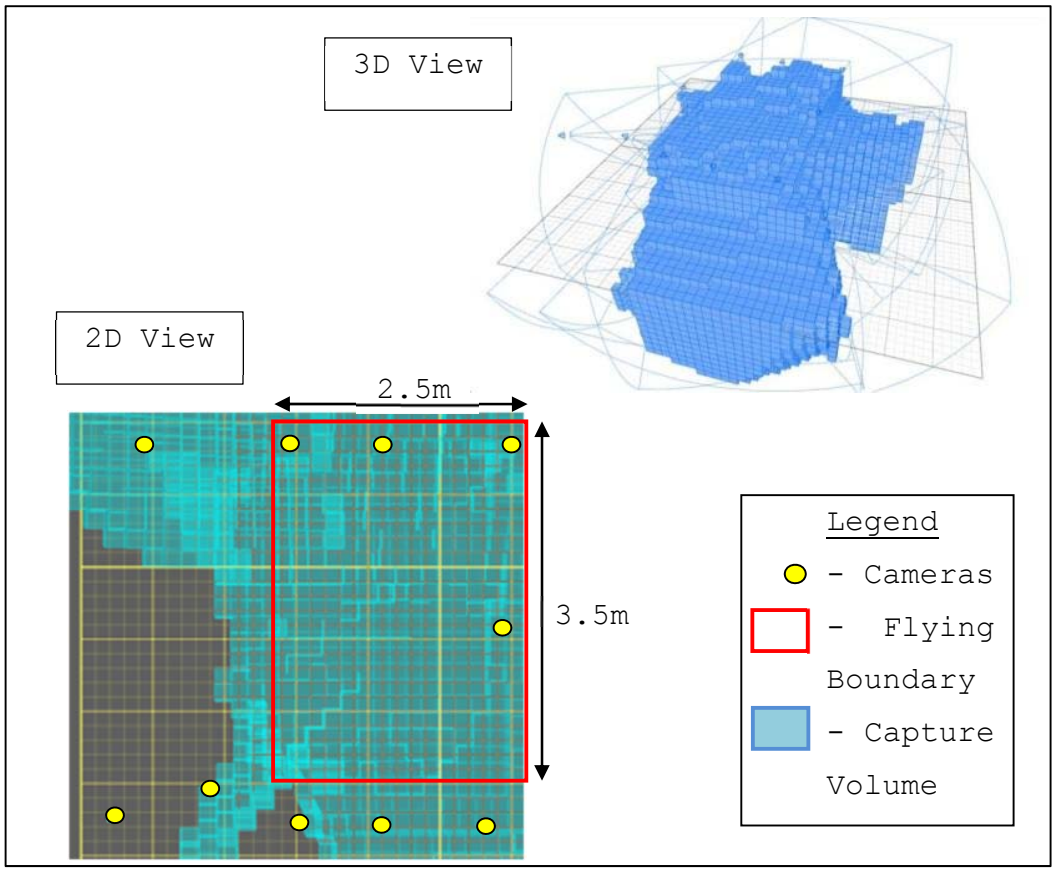


Figure 28. Capture volume from Optitrack system

The capture volume reflects the ability of locating the UAV position with 3 cameras. This constrains the flying boundary of the UAV into the rectangular box of 2.5m by 3.5m as shown in Figure 28. This flying boundary will be incorporated in the penalty function for the implementation of the direct method.

3. Functional flow

The architecture design of the Q-Ball control environment is performed using CORE software to generate the functional flow of the system, as well as other operational and system views. Since the system is designed

for use in student experiments, its architectural design is based on these experimental needs. Figure 29 shows the physical decomposition or Systems View 4a (Systems Functionality View) of the lab setup. The system consists of six primary system components: QBall UAV, control station, Optitrack system, battery charging station, tools, and landing mat. Each of the system components is further decomposed into their subsystem components.

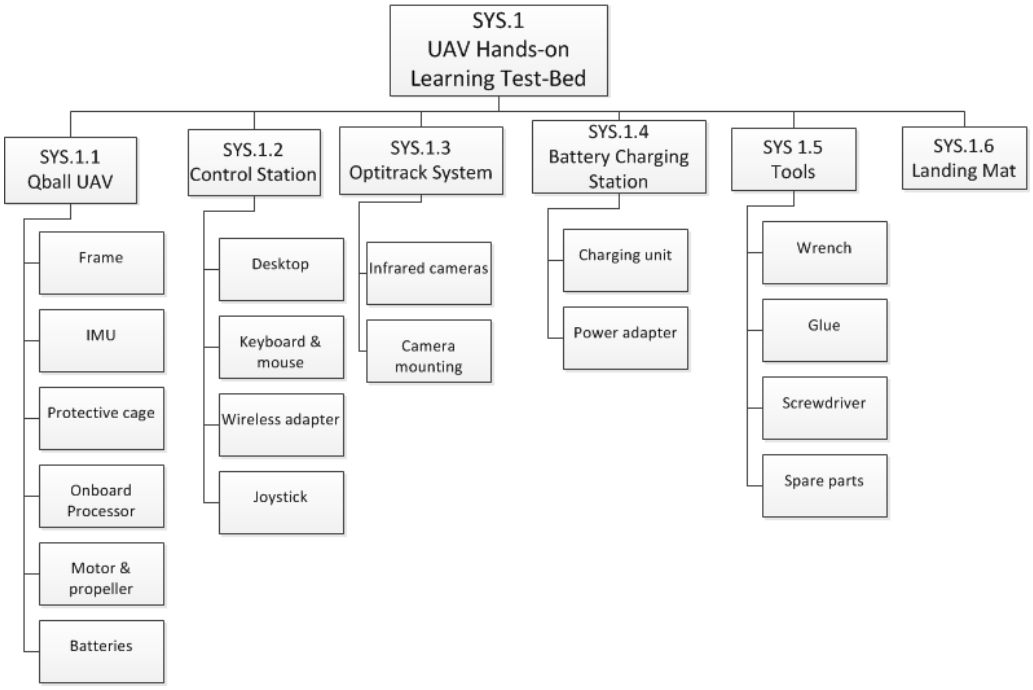


Figure 29. Physical decomposition of the lab setup

An enhanced functional flow block diagram (EFFBD) of the operation of the system is shown in Figure 30–31. The first process is to charge up the batteries with the charging station. This is followed by starting up the Optitrack system. The flow is then broken into a path for each of the three main systems: the QBall UAV, the Control Station, and the Optitrack System. The process then

involves starting the system, performing communication between these systems, and then the flight of the UAV. Lastly, the data analysis is carried out by extracting the flight data and shutting down the system.

The procedures for starting up the systems and downloading the software to the UAVs and setting up the multiple UAVs are created specifically for the use in the Bullard lab. It can be found in the Appendix under *Procedures for starting up QBall System in Bullard Lab*.

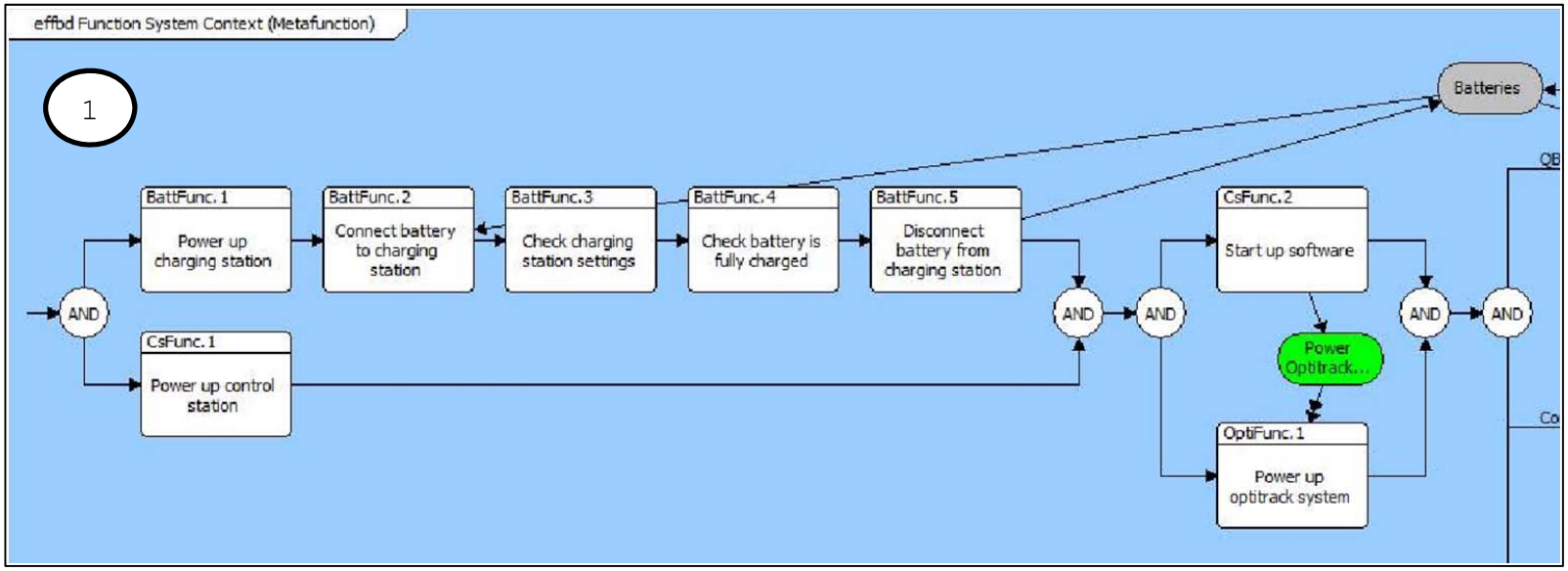


Figure 30. EFFBD of operating QBall UAV, view 1

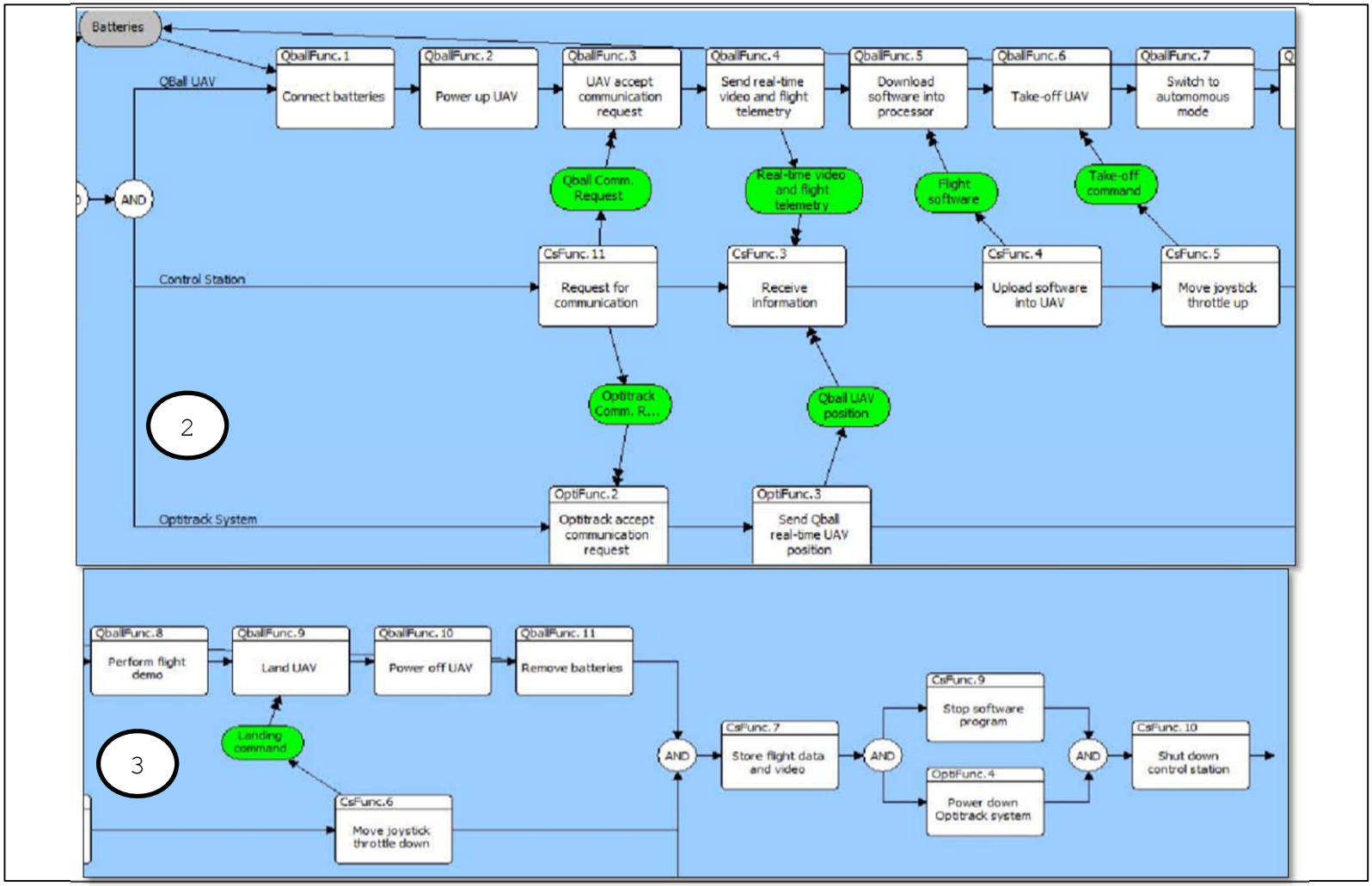


Figure 31. EFFBD of operating QBall UAV, view 2 and 3

IV. DIRECT METHOD BASED CONTROL STRATEGY

A. INTRODUCTION

There is an increasing need to reduce human involvement in the control of unmanned systems in order to minimize operator fatigue and error in the field. In an urban operation, the need for human intervention increases since way-point navigation does not work and the only option available is manual control [33]. This method of control requires more than three persons to operate one vehicle, due to the limited LOS and dynamic obstacles of the environment. In order to achieve full autonomy, a controller must generate optimal or near-optimal trajectory to perform this type of mission. There are several well-known optimization software packages, such as: OTIS, SOCS, DIRCOL, and DIDO. References [34], [35], [36] and [37] suggested the problem could be solved relatively quickly, but the solution involves hundreds and thousands of varied parameters. Therefore, an optimal real-time solution may not be possible. There is a need to simplify the problem or use numerical algorithms to provide near-optimal rather than optimal solutions in real time [6]. Direct methods have been used since the 60s using Professor Tarenko's ideas, whose research helped engineers develop algorithms real-time and on-board for near-optimal (quasi-optimal) trajectories for combat vehicles and missiles [38], [39], [40].

In this paper, the proposed implementation method is the direct method of calculus of variations in exploiting the inverse dynamics of a vehicle in the virtual domain

(IDVD). This method is based on inverting the dynamics by making use of the differential flatness (a property of a system), and derives a set of parameters to control the vehicle using the virtual domain [24]. It only requires a few varied parameters and minimal computational power to generate quasi-optimal trajectories capable of respecting the vehicle constraints as well as avoiding collisions. Figure 32 illustrates the flow of the direct method in generating the trajectory.

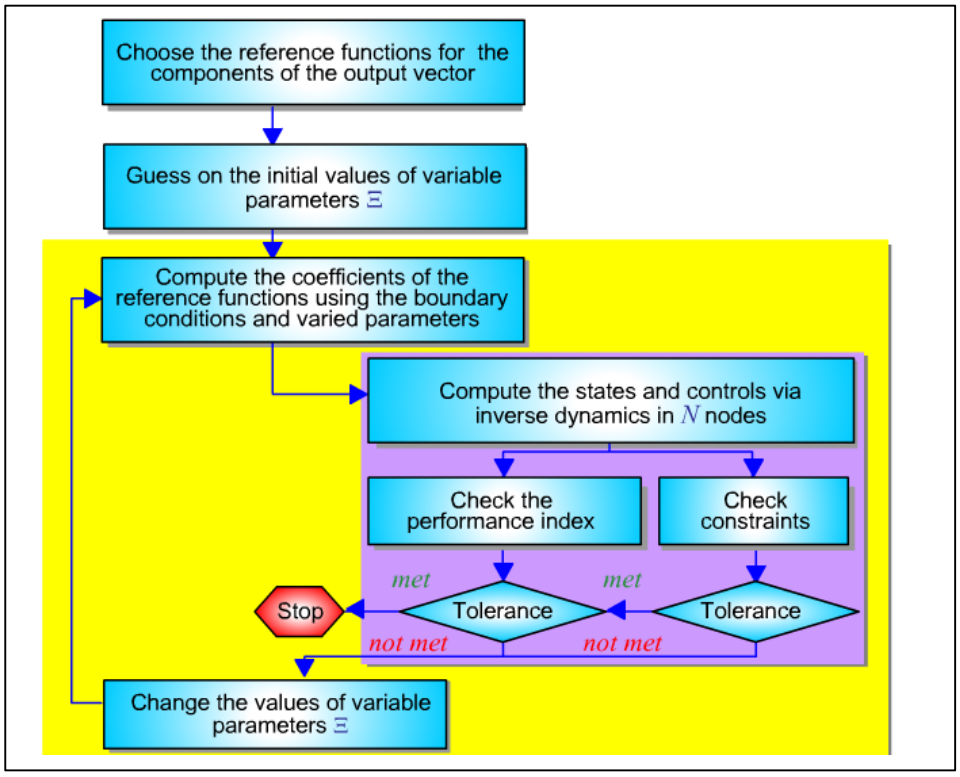


Figure 32. Direct method flow. From [41].

The following sections will focus on the key steps in the process flow for developing the collision-free trajectories.

B. ARCHITECTURE OF CONTROLLER

Figure 33 shows the general architecture of the suggested control. The architecture of the controller consists of two main loops. The method allows real-time trajectory regeneration during the mission. This enables the UAVs' control to be more robust and adapt to different scenarios. For instance, a new quasi-optimal trajectory is generated when the mission objective changes during flight, or there are large discrepancies between the current state and the suggested path due to disturbances, or even when the UAV spots an obstacle with its camera. Depending on the mission and hardware on-board, this trajectory generation loop updates every 10 to 100s.

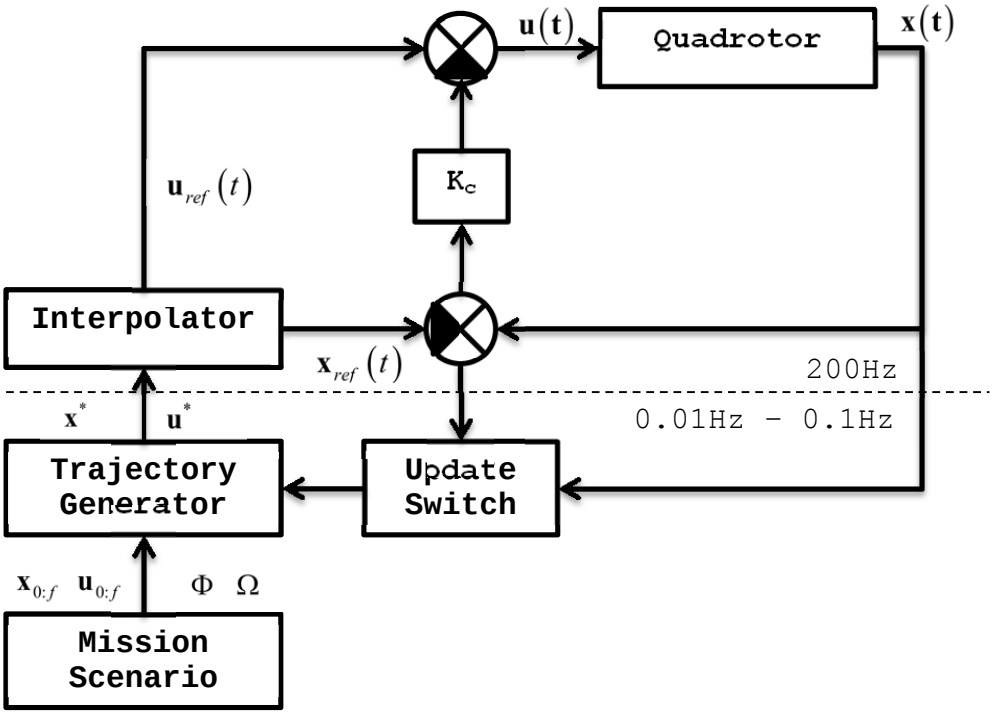


Figure 33. Architecture of direct method control for quadrotor. After [24].

The inner loop is the traditional control of the UAV; it employs the LQR controller to monitor the trajectory. The interpolator generates samples of the reference trajectory at the frequency rate and passes the commands to the controller. The LQR controller then corrects the UAV with appropriate control commands, and also counters any disturbances encountered. This loop runs with a much faster rate of 0.005s.

C. TRAJECTORY OPTIMIZATION

1. Defining a Reference Trajectory

One of the key ideas behind the inverse dynamics in the virtual domain (IDVD) method is that it allows decoupling time and space optimization by creating a reference trajectory which is independent of any time derivative constraints. This is done by employing a virtual variable "τ" as the independent variable in parameterizations as opposed to time of say a path length [41]. This variable varies between 0 and some finite value τ_f, where τ_f is considered as one of the varied parameters of the trajectory optimization problem. Once the optimal trajectory is found in the virtual domain it is then mapped from the virtual domain back to the time domain by using a variable speed factor as explained later.

Depending on a particular task (and vehicle dynamics) the IDVD method can make use of different parameterizations approximating three Cartesian coordinates of a moving object. The order of parameterization, or in other words the number of terms or coefficients is determined by the number of initial and final conditions that need to be satisfied. To be more specific, if we want to satisfy up to

the second-order derivative constraints on the both ends of a trajectory, we cannot use anything less than the 5th-order parameterization, otherwise we will simply not have enough terms. If in the latter case we choose a higher-order approximation we could use these extra terms (parameters) to expand the class of the trajectories we could choose from. In the latter example, it would be natural to increase the order of approximation to 7 and use the third-order derivatives (jerks) at the both ends as additional varied parameters.

The easiest way would be to model quadrotor's maneuver trajectory as a polynomial function. Each of the three coordinates in this case would be represented by an Nth order polynomial in the form of

$$\begin{aligned} x(\tau) &= P_x(\tau) = \sum_{i=0}^{N} a_{xi}\tau^i \\ y(\tau) &= P_y(\tau) = \sum_{i=0}^{N} a_{yi}\tau^i \\ z(\tau) &= P_z(\tau) = \sum_{i=0}^{N} a_{zi}\tau^i \end{aligned} \tag{8}$$

Following the problem formulation, the trajectory has to have smooth transition in its initial and final position, speed and controls (accelerations). Hence they are specified (given). By introducing the third-order derivative as a varied parameter N becomes equal to 7. The coefficients of a parameterization (8) then can be found from

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \tau_f & \tau_f^2 & \tau_f^3 & \tau_f^4 & \tau_f^5 & \tau_f^6 & \tau_f^7 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2\tau_f & 3\tau_f^2 & 4\tau_f^3 & 5\tau_f^4 & 6\tau_f^5 & 7\tau_f^6 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6\tau_f & 12\tau_f^2 & 20\tau_f^3 & 30\tau_f^4 & 42\tau_f^5 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 24\tau_f & 60\tau_f^2 & 120\tau_f^3 & 410\tau_f^4 \end{bmatrix} \begin{bmatrix} a_{x0} \\ a_{x1} \\ a_{x2} \\ a_{x3} \\ a_{x4} \\ a_{x5} \\ a_{x6} \\ a_{x7} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_f \\ x_0' \\ x_f' \\ x_0'' \\ x_f'' \\ x_0''' \\ x_f''' \end{bmatrix} \quad (9)$$

(Equation (9) applies similarly for y and z coordinates.)

Alternatively, a trajectory parameterization can rely on the trigonometric terms. For N=7 it becomes

$$\begin{aligned} x(\bar{\tau}) &= P_x(\bar{\tau}) = a_{x0} + \sum_{i=1}^{3} a_{xi}\cos(i\pi\bar{\tau}) + \sum_{i=1}^{4} b_{xi}\sin(i\pi\bar{\tau}) \\ y(\bar{\tau}) &= P_y(\bar{\tau}) = a_{y0} + \sum_{i=1}^{3} a_{yi}\cos(i\pi\bar{\tau}) + \sum_{i=1}^{4} b_{yi}\sin(i\pi\bar{\tau}) \\ z(\bar{\tau}) &= P_z(\bar{\tau}) = a_{z0} + \sum_{i=1}^{3} a_{zi}\cos(i\pi\bar{\tau}) + \sum_{i=1}^{4} b_{zi}\sin(i\pi\bar{\tau}) \end{aligned} \quad (10)$$

(here $\bar{\tau} = \tau\tau_f^{-1}$). In this case the unknown coefficients of (10) will be found from resolving the following set of algebraic equations

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \pi & 2\pi & 3\pi & 4\pi \\ 0 & 0 & 0 & 0 & -\pi & 2\pi & -3\pi & 4\pi \\ 0 & -\pi^2 & -(2\pi)^2 & -(3\pi)^2 & 0 & 0 & 0 & 0 \\ 0 & \pi^2 & -(2\pi)^2 & (3\pi)^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\pi^3 & -(2\pi)^3 & -(3\pi)^3 & -(4\pi)^3 \\ 0 & 0 & 0 & 0 & \pi^3 & -(2\pi)^3 & (3\pi)^3 & -(4\pi)^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_f \\ x_0'\tau_f \\ x_f'\tau_f \\ x_0''\tau_f^2 \\ x_f''\tau_f^2 \\ x_0'''\tau_f^3 \\ x_f'''\tau_f^3 \end{bmatrix} \quad (11)$$

Combining (8) and (10) yields even a better (for our particular case) parameterization that benefits from both monomial and trigonometric terms

$$\begin{aligned} x(\bar{\tau}) &= P_x(\bar{\tau}) = a_{x0} + a_{x1}\bar{\tau} + a_{x2}\bar{\tau}^2 + a_{x3}\bar{\tau}^3 + \ldots \\ &\quad b_{x1}\sin(\pi\bar{\tau}) + b_{x2}\sin(2\pi\bar{\tau}) + b_{x3}\sin(3\pi\bar{\tau}) + b_{x4}\sin(4\pi\bar{\tau}) \\ y(\bar{\tau}) &= P_y(\bar{\tau}) = a_{y0} + a_{y1}\bar{\tau} + a_{y2}\bar{\tau}^2 + a_{y3}\bar{\tau}^3 + \ldots \\ &\quad b_{y1}\sin(\pi\bar{\tau}) + b_{y2}\sin(2\pi\bar{\tau}) + b_{y3}\sin(3\pi\bar{\tau}) + b_{y4}\sin(4\pi\bar{\tau}) \\ z(\bar{\tau}) &= P_z(\bar{\tau}) = a_{z0} + a_{z1}\bar{\tau} + a_{z2}\bar{\tau}^2 + a_{z3}\bar{\tau}^3 + \ldots \\ &\quad b_{z1}\sin(\pi\bar{\tau}) + b_{z2}\sin(2\pi\bar{\tau}) + b_{z3}\sin(3\pi\bar{\tau}) + b_{z4}\sin(4\pi\bar{\tau}) \end{aligned} \tag{12}$$

The coefficients in this case are determined from

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \pi & 2\pi & 3\pi & 4\pi \\ 0 & 1 & 2 & 3 & -\pi & 2\pi & -3\pi & 4\pi \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & -\pi^3 & -(2\pi)^3 & -(3\pi)^3 & -(4\pi)^3 \\ 0 & 0 & 0 & 6 & \pi^3 & -(2\pi)^3 & (3\pi)^3 & -(4\pi)^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_f \\ x_0'\tau_f \\ x_f'\tau_f \\ x_0''\tau_f^2 \\ x_f''\tau_f^2 \\ x_0'''\tau_f^3 \\ x_f'''\tau_f^3 \end{bmatrix} \tag{13}$$

2. Time and Space Decoupling

As shown in [41], using time as an independent variable leads to a disaster:

$$V(t) = \sqrt{\dot{x}^2(t) + \dot{y}^2(t) + \dot{z}^2(t)} = \sqrt{\dot{P}_x^2(t) + \dot{P}_y^2(t) + \dot{P}_z^2} \tag{14}$$

This means that each candidate trajectory has a unique unalterable speed profile.

In order to be able to vary a speed profile along a predetermined path, i.e., decouple the trajectory from the

speed profile, an argument τ must be used to later connect to time through the speed factor:

$$\lambda(\tau)=\frac{d\tau}{dt} \tag{15}$$

By utilizing this speed factor, the speed profile can be varied along the same trajectory by changing the speed factor [41]:

$$V(\tau)=\lambda(\tau)\sqrt{P_x'^2(\tau)+P_y'^2(\tau)+P_z'^2(\tau)} \tag{16}$$

We may have the initial and final value of λ set to one (it simply rescales the virtual arc length τ_f) and the 1st order derivative set to zero (for smooth departure and arrival). Then, following the previous discussion, to increase the flexibility of the speed reference profile, the 2nd order derivatives of the speed factor can be used as extra varied parameters. This requires a 5th-order parameterization. Following [6] and employing a polynomial of the form of (8) we resolve for the corresponding coefficients utilizing algebraic equations, similar to those of (9):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{6}\tau_f^3 & \frac{1}{12}\tau_f^4 & \frac{1}{20}\tau_f^5 \\ 0 & 1 & \tau_f & \frac{1}{2}\tau_f^2 & \frac{1}{3}\tau_f^3 & \frac{1}{4}\tau_f^4 \\ 0 & 0 & 1 & \tau_f & \tau_f^2 & \tau_f^3 \end{bmatrix} \begin{bmatrix} a_{\lambda 0} \\ a_{\lambda 1} \\ a_{\lambda 2} \\ a_{\lambda 3} \\ a_{\lambda 4} \\ a_{\lambda 5} \end{bmatrix} = \begin{bmatrix} \lambda_0 \\ \lambda'_0 \\ \lambda''_0 \\ \lambda'''_0 \\ \lambda_f \\ \lambda'_f \\ \lambda''_f \\ \lambda'''_f \end{bmatrix} \tag{17}$$

Thus the speed profile can be computed as:

$$V(\tau) = P_{\lambda}(\tau)\sqrt{P_x'^2(\tau) + P_y'^2(\tau) + P_z'^2(\tau)} \tag{18}$$

3. Inverse Dynamics

In order to determine the controls for the vehicle, inverse dynamics of the system are needed. By using the differential flatness property of the system, the inverse dynamics of the vehicle can be derived. From the definition of [42], differential flatness is the property of a system, such that all of its states and controls can be expressed in terms of the output vector and its derivatives.

The state vector can be expressed as a function of the output vector, and its derivatives as in [24]:

$$\theta = \arctan\left(\frac{\ddot{x}}{g - \ddot{z}}\right) \tag{19}$$

$$\phi = \arcsin\left(\frac{-\ddot{y}}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + \left(g - \ddot{z}\right)^2}}\right) \tag{20}$$

The derivatives of Eqs. 19 and 20 yield:

$$\dot{\theta} = \frac{x\left(g - \ddot{z}\right) + \ddot{x} \cdot z}{\left(g - \ddot{z}\right)^2 + \ddot{x}^2} \tag{21}$$

$$\dot{\phi} = \frac{\left(\ddot{x}\dddot{x} - \left(g - \ddot{z}\right)\dddot{z}\right)\ddot{y} - \left(\ddot{x}^2 + \left(g - \ddot{z}\right)^2\right)\dddot{y}}{\left(\ddot{x}^2 + \ddot{y}^2 + \left(g - \ddot{z}\right)^2\right)\sqrt{\ddot{x}^2 + \left(g - \ddot{z}\right)^2}} \tag{22}$$

It should be noted that despite of the fact that the model is developed for three-dimensional scenarios, due to a limited operational area (low ceiling), we enforce a vertical coordinate to be constant (utilizing a separate

altitude-hold controller) and therefore the vertical coordinate and its derivatives play no role in further simulations.

4. Cost Function

The cost function is a quantitative measure of the optimality of the trajectory [24]. It is the sum of the running costs of not meeting the constraints. From the perspective of a single quadrotor, the key constraints are arrival times, vehicle constraints (which are the maximum roll and pitch angles), and the obstacle constraints. Based on these constraints, the cost function J was derived as shown:

$$
\begin{aligned}
J = {} & w_1 \left(\frac{\left(t_{desired} - t_{end}\right)^2}{t_{desired}^2} \right) \\
& + w_2 \left(\max\left(0, \frac{\phi_{\max} - \phi_{threshold}}{\phi_{threshold}}\right)^2 + \max\left(0, \frac{\theta_{\max} - \theta_{threshold}}{\theta_{threshold}}\right)^2 \right) \\
& + w_3 \max\left(0, \frac{d_{threshold,Obs} - d_{\min,Obs}}{d_{threshold,Obs}}\right)^2 \\
& + w_4 \left(\begin{array}{l} \max\left(0, \frac{|X_{\min}| - X_{threshold}}{X_{threshold}}\right)^2 + \max\left(0, \frac{|X_{\max}| - X_{threshold}}{X_{threshold}}\right)^2 \\ + \max\left(0, \frac{|Y_{\min}| - Y_{threshold}}{Y_{threshold}}\right)^2 + \max\left(0, \frac{|Y_{\max}| - Y_{threshold}}{Y_{threshold}}\right)^2 \end{array} \right)
\end{aligned}
\tag{23}
$$

where w_1, w_2, w_3 and w_4 are weighting factors that can be tuned to control for each individual penalty, and where $t_{desired}$, t_{end}, ϕ_{max}, $\phi_{threshold}$, θ_{max}, $\theta_{threshold}$, $d_{threshold,Obs}$, $d_{min,Obs}$ are, respectively: desired arrival time entered by the user, end time of the maneuver, maximum roll in the

maneuver, roll limit of the controller, maximum pitch in the maneuver, pitch limit of the controller, allowable distance from the obstacle, and the minimum distance from the obstacle in the maneuver.

In the case of multiple UAVs, the same cost function, augmented with an additional constraint of keeping spatial separation between them, can be used to generate a trajectory for each UAV. In practice such a trajectory will be generated onboard of each UAV, i.e., in a decentralized manner. In the lab implementation we chose to do a centralized trajectory optimization, i.e., producing trajectories for all UAVs simultaneously within a single optimization routine. Due to the limited space available in the lab, we only consider two UAVs operating at the same height above the floor. The combined cost function J for both UAVs (UAV A and UAV B) is then as shown:

$$
\begin{aligned}
J = {} & w_1\left(\frac{\left(t_{desired}-t_{end,UAVA}\right)^2}{t_{desired}^2}+\frac{\left(t_{desired}-t_{end,UAVB}\right)^2}{t_{desired}^2}\right) \\
& +w_2\left(\begin{array}{l}\max\left(0,\dfrac{\phi_{\max,UAVA}-\phi_{threshold}}{\phi_{threshold}}\right)^2+\max\left(0,\dfrac{\phi_{\max,UAVB}-\phi_{threshold}}{\phi_{threshold}}\right)^2 \\ +\max\left(0,\dfrac{\theta_{\max,UAVA}-\theta_{threshold}}{\theta_{threshold}}\right)^2+\max\left(0,\dfrac{\theta_{\max,UAVB}-\theta_{threshold}}{\theta_{threshold}}\right)^2\end{array}\right) \\
& +w_3\max\left(0,\frac{d_{threshold,UAVs}-d_{\min,UAVs}}{d_{threshold,UAVs}}\right)^2 \\
& +w_4\max\left(0,\frac{d_{threshold,Obs}-d_{\min,Obs}}{d_{threshold,Obs}}\right)^2 \\
& +w_5\left(\begin{array}{l}\left(\begin{array}{l}\max\left(0,\dfrac{|X_{\min}|-X_{threshold}}{X_{threshold}}\right)^2+\max\left(0,\dfrac{|X_{\max}|-X_{threshold}}{X_{threshold}}\right)^2 \\ +\max\left(0,\dfrac{|Y_{\min}|-Y_{threshold}}{Y_{threshold}}\right)^2+\max\left(0,\dfrac{|Y_{\max}|-Y_{threshold}}{Y_{threshold}}\right)^2\end{array}\right)_{UAVA} \\ +\left(\begin{array}{l}\max\left(0,\dfrac{|X_{\min}|-X_{threshold}}{X_{threshold}}\right)^2+\max\left(0,\dfrac{|X_{\max}|-X_{threshold}}{X_{threshold}}\right)^2 \\ +\max\left(0,\dfrac{|Y_{\min}|-Y_{threshold}}{Y_{threshold}}\right)^2+\max\left(0,\dfrac{|Y_{\max}|-Y_{threshold}}{Y_{threshold}}\right)^2\end{array}\right)_{UAVB}\end{array}\right)
\end{aligned}
\tag{24}
$$

V. RESEARCH SCENARIO AND EXPERIMENT RESULTS

A. INTRODUCTION

The direct method, as discussed in the previous section, was validated and verified with a series of simulation tests and experiments run in the lab. This section shall discuss the results from the simulation runs and the actual experiment trials in the lab.

B. APPROACH

Due to the limited area and duration of the flight, the trajectory is computed only once on the ground station and fed to the quadrotors through the wireless connection. Prior to the actual experiment, the direct method simulation model was conducted to check the varied parameters and the states computed. This check ensures that all the constraints (the flight time, attitude limits and collision distance) are within the intended design. All the simulation runs and computation of the actual flight trajectories are performed on a desktop PC with an Intel Core i7 2.79 GHz processor and 8GB of RAM running Matlab version 7.13.0.564 (2011b) and QUARC blockset version 2.2.1.

The candidate trajectory generation model based on the algorithms described in Section IVc was developed in the Simulink modeling environment. This model was then called by the optimization routine at each iteration. The unconstrained gradient-based function fminunc was used to perform optimization. Obviously, such a design is not optimal from the standpoint of minimizing a CPU time required to find an optimal trajectory (besides computation

relies on the interpretative environment of MATLAB anyway). However, from the educational standpoint this is a good design because it allows student with little programming skills to quickly modify the trajectory optimization problem to fit his/her mission objectives. For the real-time implementation the optimization piece should be converted to executable code and incorporated into the main Qball control routine.

The same constraints are applied for all of the scenarios as follows:

- Maximum roll and pitch angle < +/- 0.2 radians
- Distance from obstacle > 0.8 m, where the obstacle has a radius of 0.2m. 0.5 m is the clearance for the radius of the quadrotor and 0.1 is a safety clearance for disturbances.
- Flight boundary is constrained by the space of the lab which is given as (-1.5 < X < 2) m and (-1.5 < Y < 1) m.

Due to the limitation in space in the laboratory, the full scenario was not able to be performed. The full scenario was broken into two portions to demonstrate the capability with Scenario 1 on single UAV avoiding an obstacle and Scenario 2 on two UAVs avoiding themselves as well as an obstacle.

C. SCENARIO 1 - SINGLE UAV MISSION

1. Scenario Parameters

This scenario illustrates a single UAV flight where there is an obstacle in its flight path. With the initial

and final conditions and the desired time of flight input into the direct method model, a quasi-optimal trajectory was generated that routes the UAV to avoid the obstacle. The initial and terminal boundary conditions are:

$$\begin{aligned} x(t_0) &= 1.75m & x(t_f) &= -0.75m \\ y(t_0) &= -0.25m & y(t_f) &= -0.25m \\ \dot{x}(t_0) &= 0 & \dot{x}(t_f) &= 0 \\ \dot{y}(t_0) &= 0 & \dot{y}(t_f) &= 0 \\ \ddot{x}(t_0) &= 0 & \ddot{x}(t_f) &= 0 \\ \ddot{y}(t_0) &= 0 & \ddot{y}(t_f) &= 0 \end{aligned} \tag{25}$$

Zero initial and final velocities and accelerations were used due to the fact of limited space. In actual operation, these parameters are the transiting velocities and accelerations during flight.

2. Results

The CPU elapsed time taken to generate the trajectory was 32.6354 seconds and the varied parameters computed were tabulated in Table 8.

Varied Parameter	Value
λ_i''	0.0098
λ_f''	0.0098
x_i'''	-0.01
y_i'''	-0.0131
x_f'''	-0.01
y_f'''	0.0131
τ_f	0.0304

Table 8. Varied parameters for scenario 1

Figure 34 shows a trajectory plot from the simulation results. The UAV was flying from the bottom of the area to the top and avoiding the obstacle at the center. The yellow circle depicts the avoidance boundary for the UAV and the red square is the obstacle. Figure 35 illustrates the speed profile of the UAV.

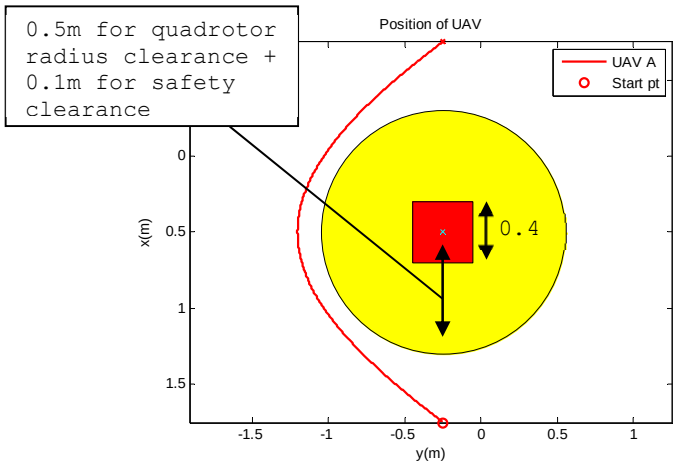


Figure 34. Trajectory from simulation result for scenario 1

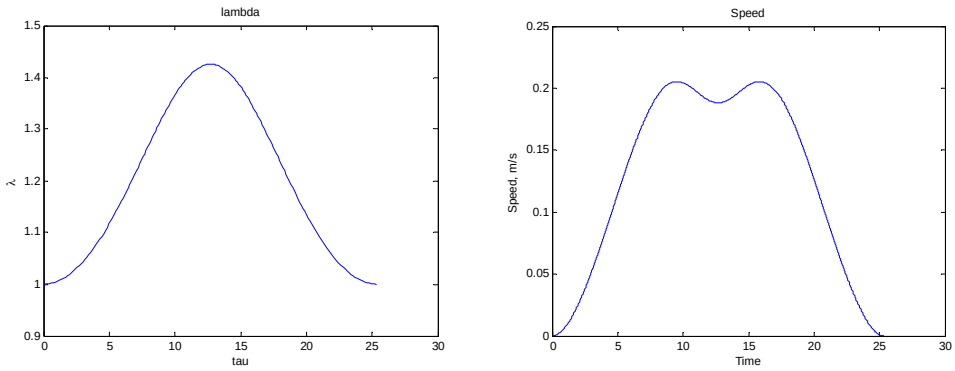


Figure 35. Speed factor(lambda) and speed for scenario 1

The actual experiment was conducted using a QBall UAV and the results for the trajectories, Euler angles and velocities were plotted as shown in Figure 36 and 37. Previous experiments performed with a correction on the Optitrack feedback with pitch and roll compensation causes a bias to the UAV flight profile. By modifying the codes to take in feedback directly from Optitrack readings improved the path following control.

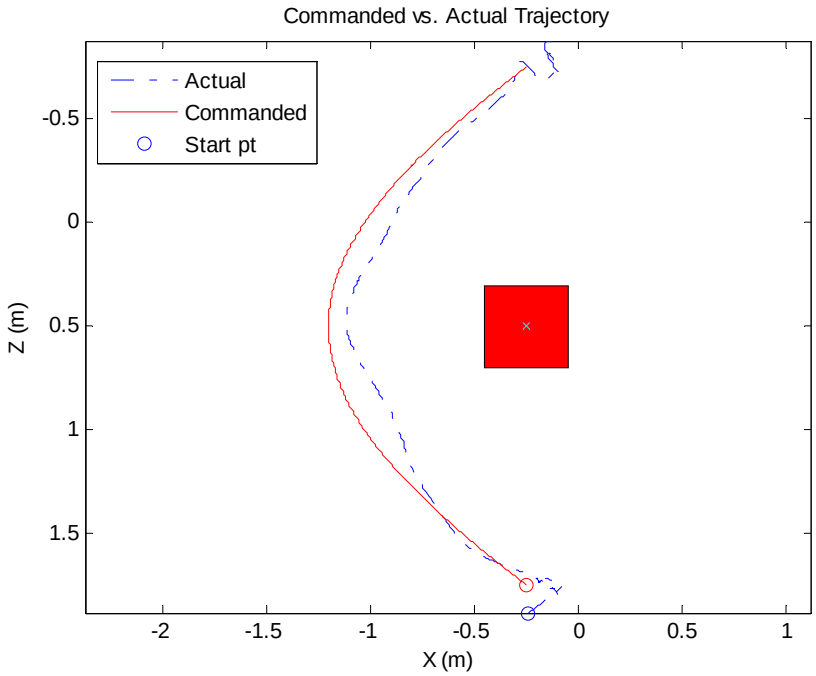


Figure 36. Trajectory from actual flight result for scenario 1

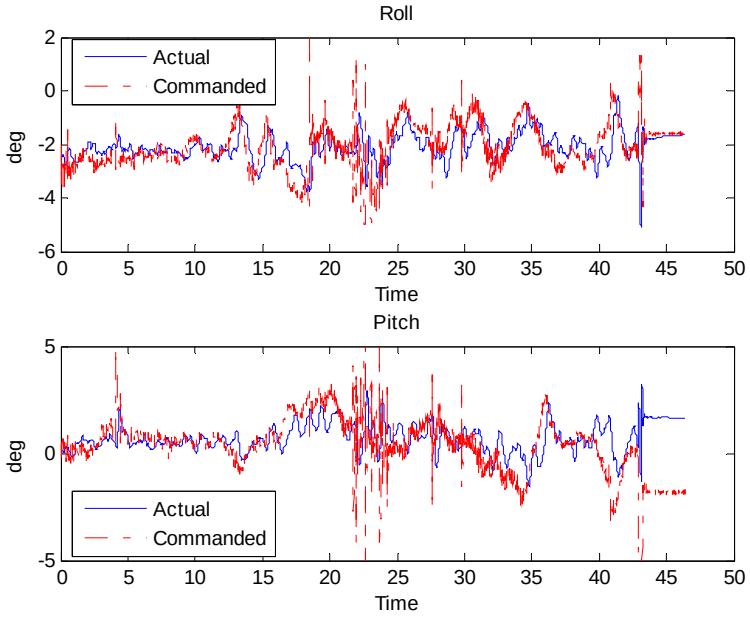


Figure 37. Euler angles from actual flight result for scenario 1

As evident from the results, the feasible path generated by the direct method proved to be collision free. A safety margin was needed to allow for disturbances that may experience during the flight.

D. SCENARIO 2 – MULTIPLE UAVS MISSION

1. Scenario Parameters

In this scenario, two UAVs (UAV A and UAV B) were used to illustrate a mission where both UAVs have to fly past each other, and at the same time, avoid an obstacle. This tests the algorithm with regard to the ability to create a trajectory that is able to generate a feasible path, and yet does not take too much time and computing power. The initial and final terminal boundary conditions are as follows:

$$\begin{array}{llll} x_A(t_0)=2m & x_A(t_f)=-1m & x_B(t_0)=-1 & x_B(t_f)=2m \\ y_A(t_0)=-0.25m & y_A(t_f)=-0.25m & y_B(t_0)=-0.25m & y_B(t_f)=-0.25m \\ \dot{x}_A(t_0)=0 & \dot{x}_A(t_f)=0 & \dot{x}_B(t_0)=0 & \dot{x}_B(t_f)=0 \\ \dot{y}_A(t_0)=0 & \dot{y}_A(t_f)=0 & \dot{y}_B(t_0)=0 & \dot{y}_B(t_f)=0 \\ \ddot{x}_A(t_0)=0 & \ddot{x}_A(t_f)=0 & \ddot{x}_B(t_0)=0 & \ddot{x}_B(t_f)=0 \\ \ddot{y}_A(t_0)=0 & \ddot{y}_A(t_f)=0 & \ddot{y}_B(t_0)=0 & \ddot{y}_B(t_f)=0 \end{array} \quad (26)$$

2. Results

The CPU elapsed time for generating the trajectory was 76.9085 seconds. The varied parameters computed are tabulated in Table 9.

UAV A - Varied Parameter	Value	UAV B - Varied Parameter	Value
$\lambda''_{i,A}$	0.01	$\lambda''_{i,B}$	0.01
$\lambda''_{f,A}$	0.01	$\lambda''_{f,B}$	0.01
$x'''_{i,A}$	-0.015	$x'''_{i,B}$	0.015
$y'''_{i,A}$	0.0107	$y'''_{i,B}$	-0.0107
$x'''_{f,A}$	-0.015	$x'''_{f,B}$	0.015
$y'''_{f,A}$	-0.0107	$y'''_{f,B}$	0.0107
$\tau_{f,A}$	0.0316	$\tau_{f,B}$	0.0316

Table 9. Varied parameters for scenario 2

As shown by the simulation results in Figures 38–39, the algorithm was able to generate a feasible path which altered the trajectory to avoid the obstacle and prevented both UAVs from colliding into each other.

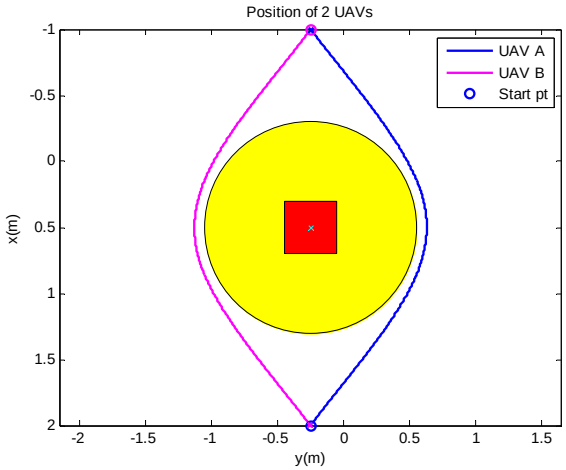


Figure 38. Trajectory from simulation result for scenario 2

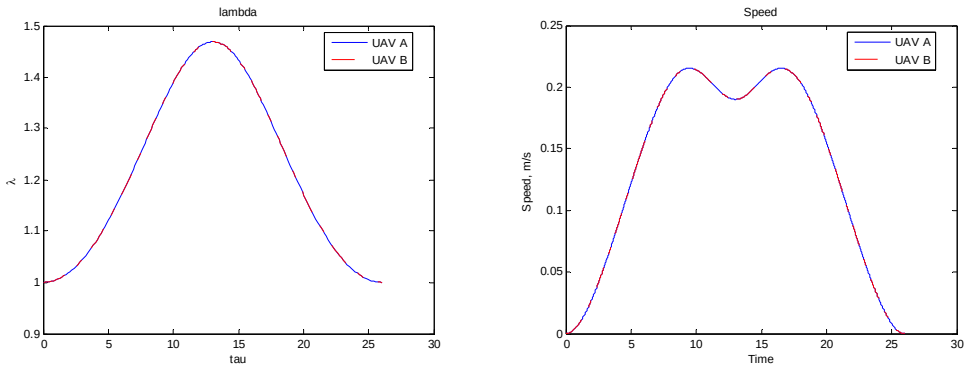


Figure 39. Speed factor(lambda) and speed for scenario 2

This trajectory was tested in an actual flight experiment with two QBall UAVs (UAV A and UAV B) and the results are plotted as shown in Figure 40 – 42.

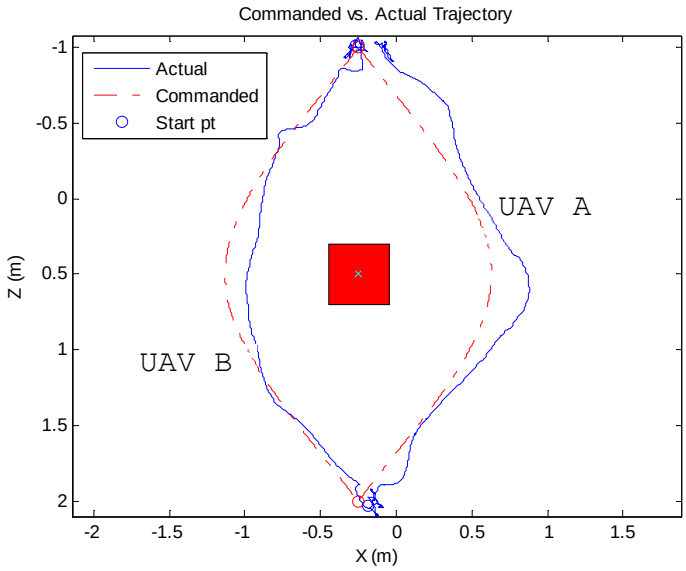


Figure 40. Trajectory from actual flight result for scenario 2

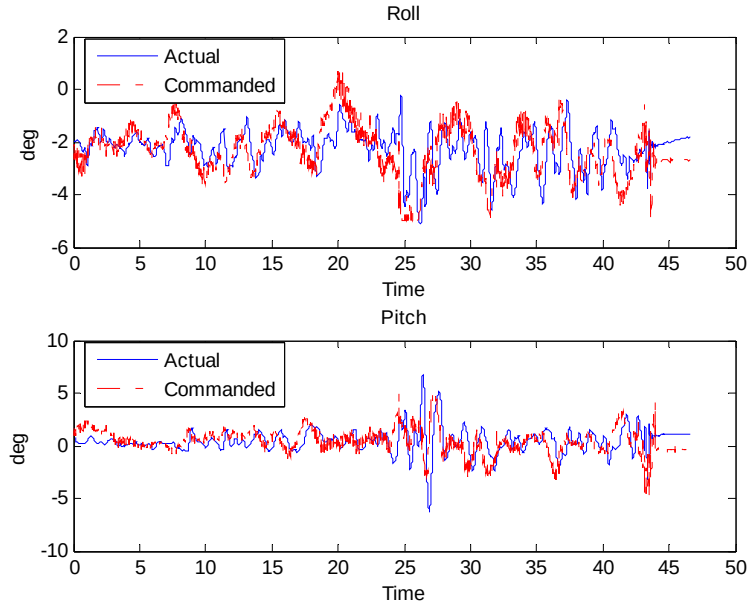


Figure 41. UAV A euler angles from actual flight result for scenario 2

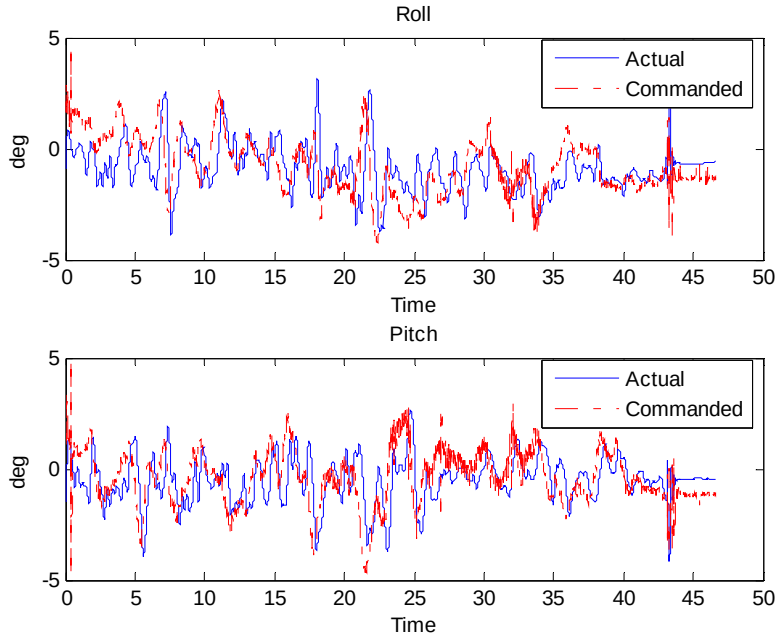


Figure 42. UAV B euler angles from actual flight result for scenario 2

From the results, both UAVs are able to avoid the obstacle and at the same time, avoid colliding into each other. Although there is some latency in the controls for both UAVs, there is a sufficient safety margin for the UAVs to fly and avoid colliding into the obstacle.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The major challenges of operating in an urban terrain, where the environment contains both dynamic obstacles and LOS issues (which increase the difficulty of operating unmanned systems), was highlighted in this paper. This paper also addressed the need to have greater autonomy so as to reduce the number of operators, as well as the importance of having unmanned systems that can carry out missions in the urban terrain.

By applying a systems engineering approach to address the problem, several solutions are recommended in this paper. A concept of operations was demonstrated in the paper to provide a high-level perspective of the system's operation of collaborative ISR missions using multiple UxVs capable of dynamic reconfigurations (required due to the complex environment of urban terrain).

By applying the direct method using IDVD, the UAVs are capable of performing non-centralized guidance and control by generating quasi-optimal trajectories for obstacle avoidance and dynamic UAV avoidance on a real-time scale. The results obtained from an indoor experiment with Quanser QBall quadrotor aircraft. These developed algorithms can now be further transferred to outdoor vehicles to be tested in a real-world environment.

B. RECOMMENDATIONS

The recommendations for future work applying for this thesis are:

- Incorporate the use of dynamic switches to call models to control multiple systems instead of opening different models.
- Experiment with multiple UAVS and UGVs in a larger scale either in an outdoor environment or in a larger laboratory.
- Use the camera to detect the distance between the UAV and the obstacle and perform the direct method of dynamic reconfiguration.
- Develop an intuitive user interface for the control station to control multiple UxVs with the use of direct methods that allow the user to choose from various feasible trajectories generated.

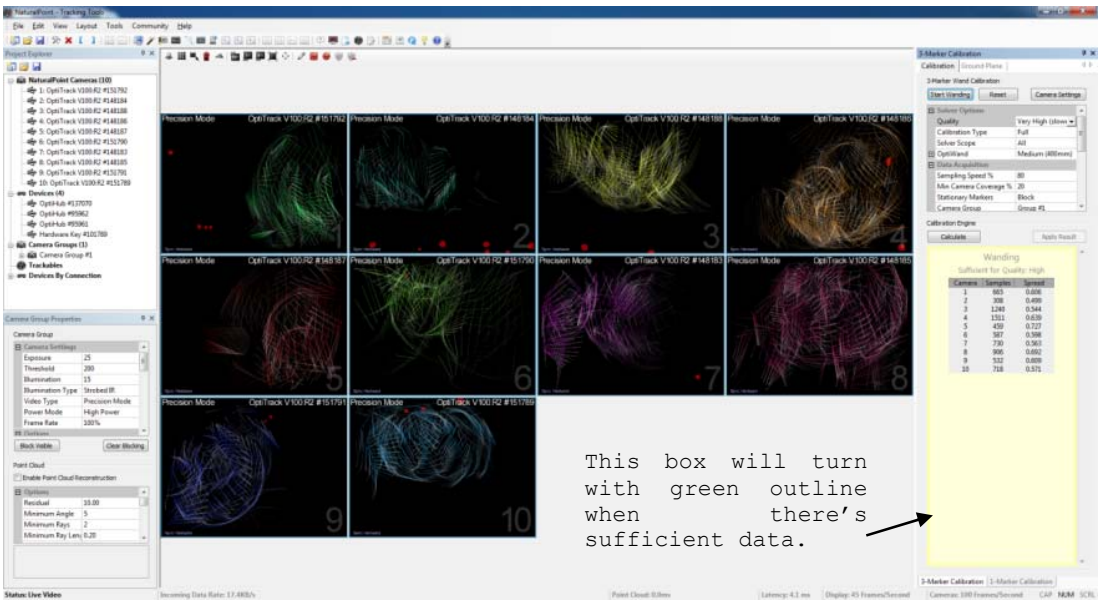
APPENDIX

Some of the images in this appendix are too wide or bleeding into the left or right margins. Ensure each image fits inside left and right margin.

A. PROCEDURE FOR OPTITRACK CAMERA SYSTEM CALIBRATION

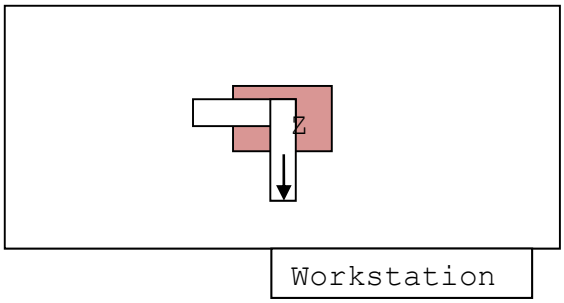
1. Open software: Desktop > Tracking Tools.
2. Click on Perform Camera Calibration.
3. Under "Solver Options" on the right-hand task pane, select Quality > Very High.
4. On the top screen, click on the icon "Block visible markers."

5. On the right-hand task pane, click on "Start Wanding." Use the wand stick and sway "Figure of 8." Make sure that all cameras have sufficient data points. The screen should look similar to the one shown below.

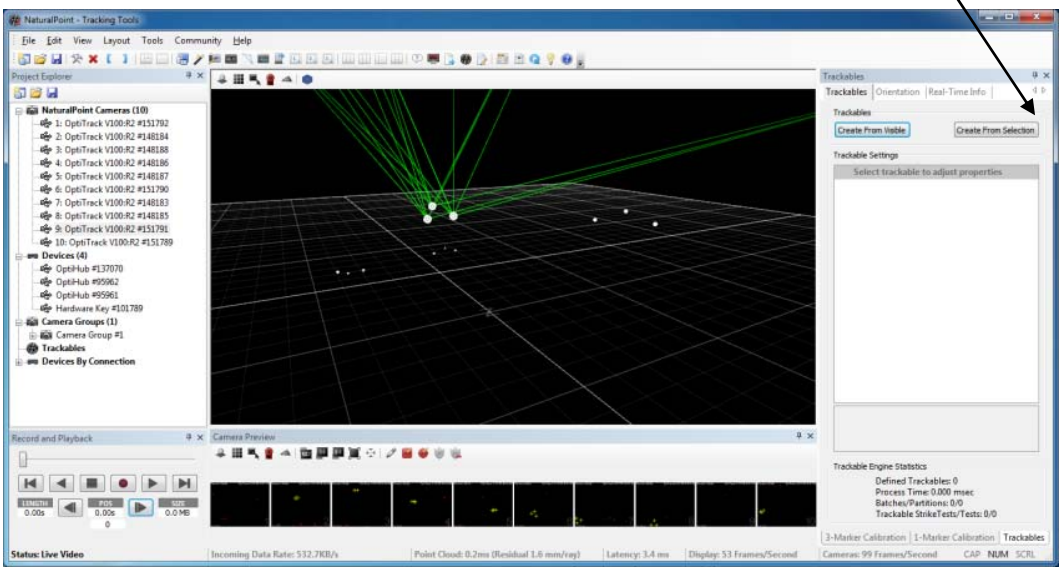


6. Click "Calculate" on right-hand task pane. Wait till the "Ready to Apply" sign appear.
7. Click "Apply Result" on right-hand task pane.

8. “Apply” and Save the file in Libraries > Documents > OptiTrack. (.cal file)
9. This will bring up the Ground Plane Calibration screen.
10. Use the L-shaped tool and set it on the pink box as shown in the figure below. Level the tool.



11. Save the file with the same name as Step 8 by clicking “Set Ground Plane.”
12. Place 12 reflectors around the area. The 12 reflectors can either four Qballs or mix of Qballs and reflector balls. This is required to gather four Qballs trackables’ locations.
13. Select three reflectors that belong to the Qball at the pink box and click on “Create from Selection.” This should form “Trackable 1.”



Remarks:

- Use the scroll wheel to zoom in and out
- Hold on to right-click to rotate the screen

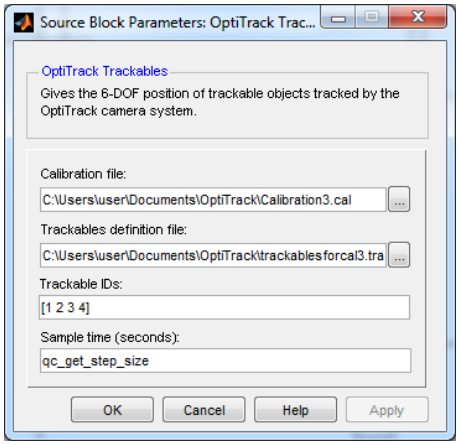
- Hold on to the scroll wheel to move around the screen

14. Repeat the same for the other reflectors. A total of four Trackables will be created.
15. Set the top reflector of the Qball as the Pivot point by left-clicking on the point. Then, right-click and "Set Trackable Pivot Point."
16. Go to File > Save Trackables.

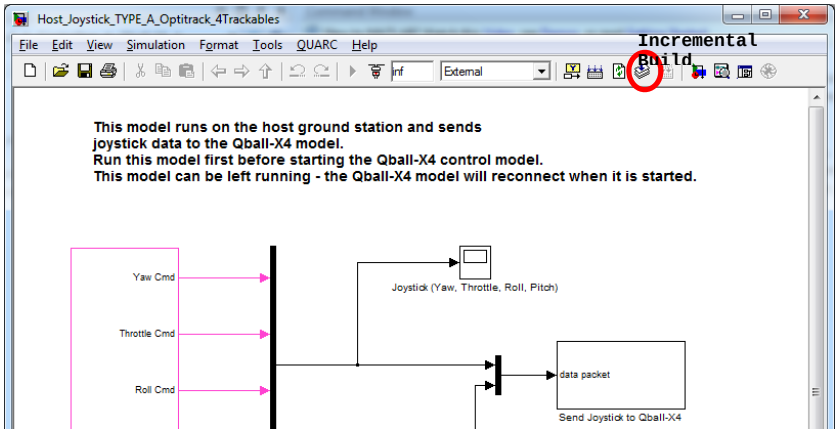
Exit the software.

B. PROCEDURE FOR STARTING UP QBALL SYSTEM IN BULLARD LAB

1. Make sure that the batteries are fully charged.
2. Strap 2 batteries on QBall.
3. Place QBall on the Pink mat with orange rod (X-axis) pointing towards the workstation.
4. Check that the wireless dongle is connected.
5. Open the models. Libraries > Documents > Chris > QBall-X4 v3 > Contollers > QBall-X4 >
   i) Host_Joystick_TYPE_A_Optitrack_4Trackables.mdl
   ii) Qball_x4_Base_v3.mdl
6. Go to Model (i), double-click the block "OptiTrack Measurements," then double-click block "OptiTrack Trackables."
7. A dialog box as shown below appears. Under Calibration File > Select the ."cal" calibration file generated from the Optitrack camera calibration performed. Repeat for Trackables definition file for ."tra" file. Details can be referred to "Procedures for Calibration Optitrack Camera.docx."

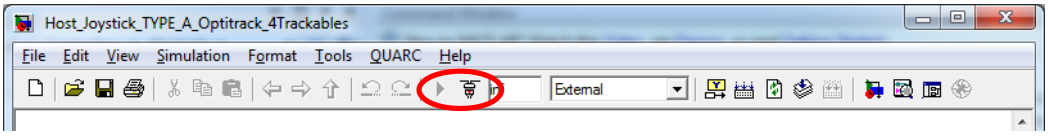


8. Connect the batteries and switch on the system. There will be consistent "Beep" sounds.
9. Go back to Model (i), click on "Incremental Build" to build the C-codes into the desktop.



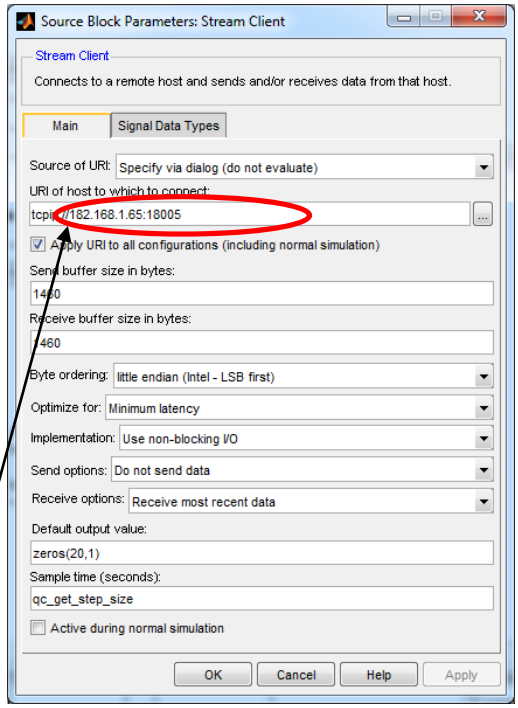
10. Click on wireless on the desktop pane and connect to the wireless network "GSAH." (May need to wait a moment for the network to appear)

11. Click on "Connect to Target" and followed by "Start real-time code."



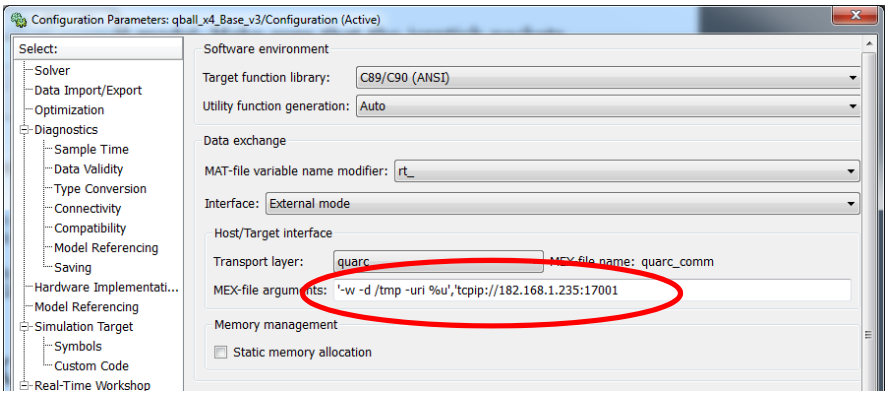
12. Check that the joystick is connected by checking on the scope for Joystick.
13. Check that the QBall is connected by moving the QBall and checking the scope for "x,y,z 1"

14. Double-click the block "Joystick from host," and then "Stream Client." A dialog box (as shown below) should appear.



15. Check that the URI tcpip address should synchronize to the computer IP address. This can be verified by Start > "Under Search: type cmd" then press "Enter." Type ipconfig and check the IPv4 Address.

16. Go to Model (ii) and go to QUARC > Options on the menu list. The dialog as shown below appear:

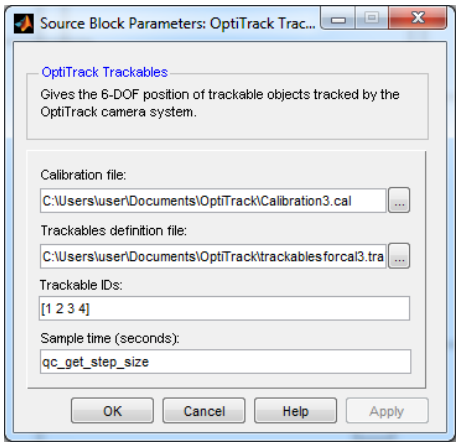


17. Check that the arguments ip address matches the QBall. There is a sticker on QBall rod to indicate the IP address of it.

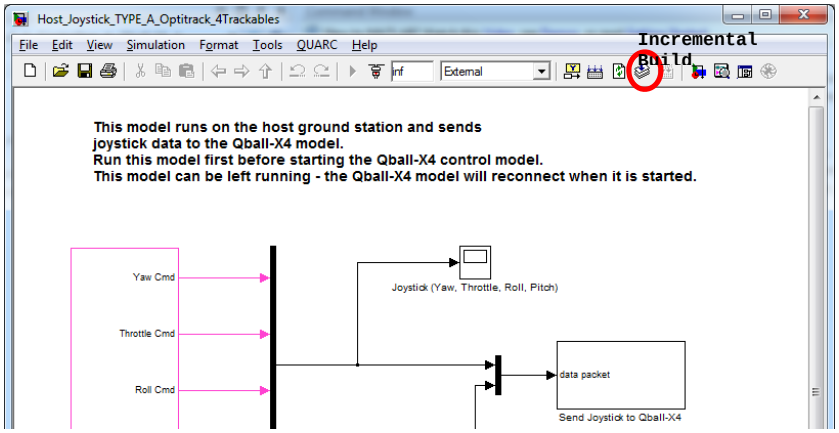
18. Click on “incremental build.” Wait for the compilation of the codes and the transferring of them to QBall. It will take about 5 minutes.
19. Click on “Connect to Target” and then “Start real-time code” to run Model (ii). The “Beep” sounds stop.
20. Start the QBall by moving the joystick’s throttle stick up to more than 50% to start the UAV.
21. UAV perform the flight plan.
22. Move the joystick’s throttle stick down to land the UAV.
23. In case of emergency, press the stop simulation button to terminate.
24. Switch off the system and unplug the batteries.
25. Charge the batteries if need be.

For Multiple UAVs run

1. Make sure that the batteries are fully charged.
2. Strap two batteries on each QBall.
3. Place QBalls on the mat with orange rod (X-axis) pointing towards the workstation.
4. Check that the wireless dongle is connected.
5. Open the models:
   i) Host_Joystick_TYPE_A_Optitrack_Trackables_UDP.mdl
   ii) qball_x4_Base_v3Qballa_pos.mdl
   iii) qball_x4_Base_v3Qballb_pos.mdl
6. Go to Model (i), double-click the block "OptiTrack Measurements," then double-click block "OptiTrack Trackables."
7. A dialog box (as shown below) appears. Under Calibration File > Select the ."cal" calibration file generated from the Optitrack camera calibration performed. Repeat for Trackables definition file for ."tra" file. Details can be referred to "Procedures for Calibration Optitrack Camera.docx."

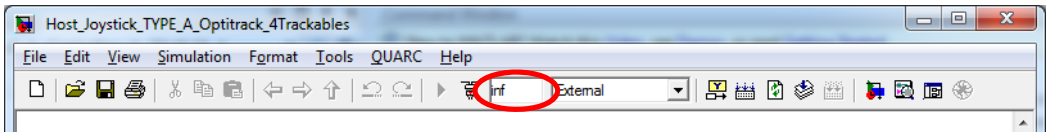


8. Connect the batteries and switch on the system. There will be consistent "Beep" sounds.
9. Go back to Model (i), click on "Incremental Build" to build the C-codes into the desktop.

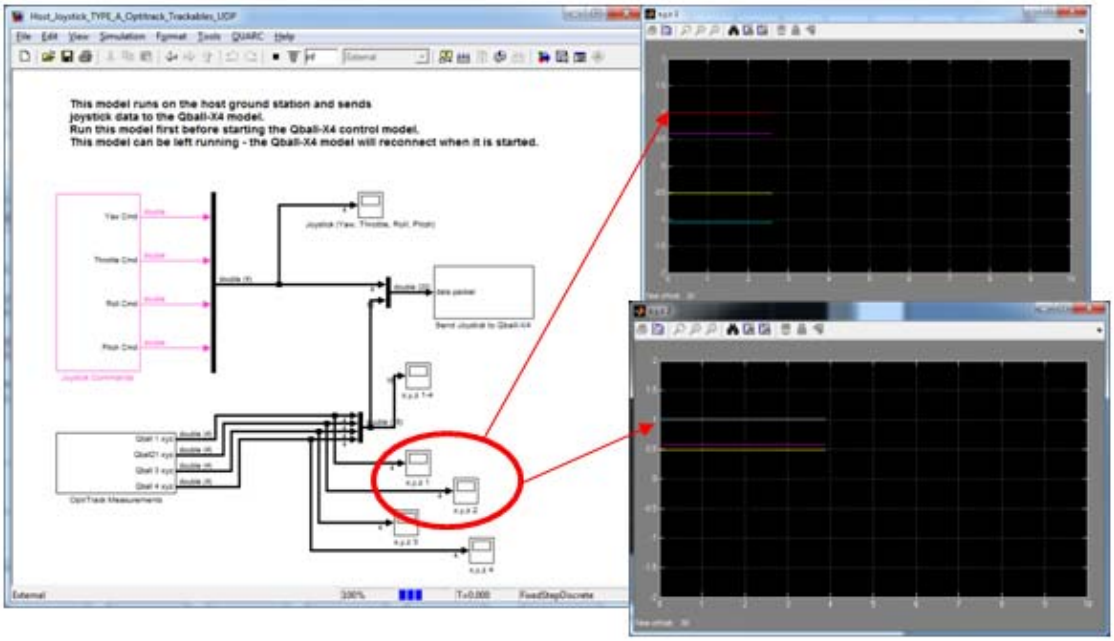


10. Click on wireless on the desktop pane and connect to the wireless network "GSAH." (May need to wait a moment for the network to appear).

11. Click on "Connect to Target" and then on "Start real-time code."



12. Check that the joystick is connected by checking on the scope for Joystick.
13. Check that the QBall is connected by moving the QBall and checking the scope for "x,y,z 1" and "x,y,z 2." The red line representing the connectivity to Optitrack system should be 1.

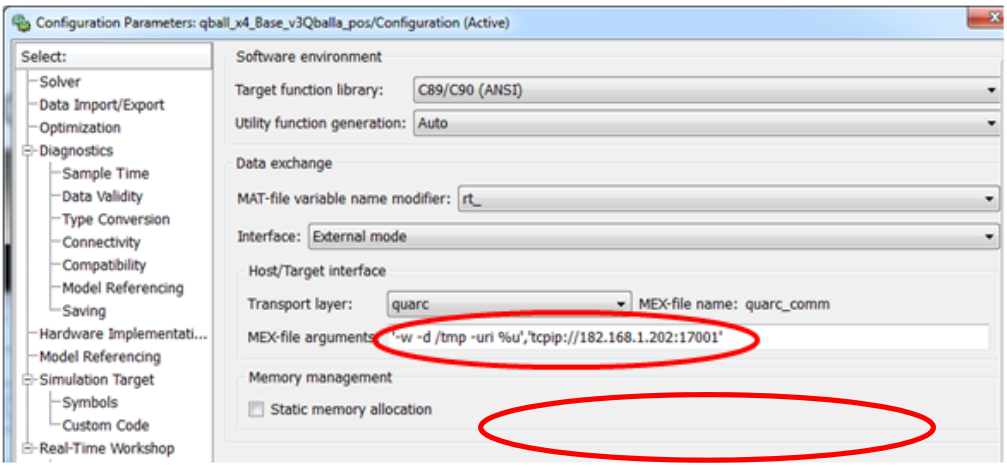


14. Double-click the block "Joystick from host" and then "Stream Client." A dialog box (as shown below) should appear.

15. Check that the URI tcpip address should synchronize to the computer IP address. This can be verified by Start > "Under Search: type cmd" then press "Enter." Type ipconfig and check the IPv4 Address.

16. Load "commands_for_scen2_.mat" from the directory.

17. On Model (ii) and (iii), go to QUARC > Options on the menu list. The dialog (as shown below) appears.



18. Check that the arguments ip address matches the QBalls. There is a sticker on QBall rod to indicate the IP address of it.
19. Click on "Incremental build" for Model (ii) and (iii). This will take about 5-10 minutes.
20. Click on "Connect to Target" for both models.
21. Before starting the codes, check that Model (i) is running. Otherwise, connect and run Model (i).
22. Click on "Start real-time code" for both models.
23. Start the QBalls by moving the joystick's throttle stick up to more than 50% to start the UAV.
24. UAV perform the flight plan.
25. Move the joystick's throttle stick down to land the UAV.
26. In case of emergency, press the stop simulation button to terminate.
27. Switch off the system and unplug the batteries.
28. Charge the batteries if need be.

LIST OF REFERENCES

[1] World Resources, *1998–99: A Guide to the Global Environment*. New York: Oxford University Press, 1998.

[2] T. Samad, J. S. Bay and D. Godbole, "Network-centric systems for military operations in urban terrain: the role of UAVs," *Proc. IEEE Proceedings of the IEEE* vol. 95, no. 1, pp. 92–107, January 2007.

[3] A. Vick et al., "Aerospace operations in urban environments: exploring new concepts," pp. 285, 2000. Available: http://www.rand.org/publications/MR/MR1187/.

[4] B. C. Chua et al., "Unmanned air and ground vehicles for military operations on urban terrain (MOUT) – technical and operational challenges," TDSI, Singapore, 2007.

[5] G. Leng and O. Yakimenko, "Situational awareness in urban areas," unpublished.

[6] G. Milionis, "A framework for collaborative quadrotor - ground robot missions," M.S. thesis, Department of Physics, Naval Postgraduate School, Monterey, CA, 2011.

[7] Aerovironment. (2012, April 7). *Qube: Public safety small UAS*. [Online]. Available: http://www.avinc.com/uas/small_uas/qube/.

[8] Middlesex University. (2012, April 7). *'Spy' drone generates worldwide interest*. [Online]. Available: http://www.mdx.ac.uk/aboutus/news-events/middlesex-matters/sq4-uav.aspx.

[9] Draganfly Innovation Inc. (2012, April 7). *Draganfly X6*. [Online]. Available: http://www.draganfly.com/uav-helicopter/draganflyer-x6/applications/military.php.

[10] Aerovironment. (2012, April 7). *UAS advanced development: Shrike VTOL*. [Online]. Available: http://www.avinc.com/uas/adc/shrikevtol/.

[11] Unmanned Editor. (2012, April 7). *Ghost – micro unmanned aerial vehicle (MUAV) specifications*. [Online]. Available: http://www.unmanned.co.uk/autonomous-unmanned-vehicles/uav-data-specifications-fact-sheets/ghost-micro-unmanned-aerial-vehicle-muav-specifications/.

[12] Aeryon Lab Inc. (2012, April 7). *Military applications*. [Online]. Available: http://www.aeryon.com/applications/military.html.

[13] W. E. Gortney. (2012, February 15). *Joint Publication 1–02. Department of Defense Dictionary of Military and Associated Terms*. [Online]. Available: http://www.dtic.mil/doctrine/dod_dictionary.

[14] S. Roberts, "Development of UAV sensors and information management," Royal United Services Institute Defence Systems, London, Great Britain, 2010.

[15] DJI. (2012, August 21). *Spreading wings S800 - technical specifications*. [Online]. Available: http://www.dji-innovations.com/products/spreading-wings-s800/specifications/.

[16] T. Nugent Jr., J. Kare, D. Bashford, C. Erickson and J. Alexander, "12-hour hover: flight demonstration of a laser-powered quadrocopter," in *AUVSI*, Kent, WA, 2011.

[17] T. J. Nugent Jr. and J. T. Kare, "Laser power beaming for defense and security applications," in *DS&S Conference*, Kent, WA, 2011.

[18] S. Jwa and U. Ozguner, "Multi-UAV sensing over urban areas via layered data fusion," in *Statistical Signal Processing, 2007. SSP '07. IEEE/SP 14th Workshop*, 2007 © IEEE. doi: 10.1109/SSP.2007.4301201.

[19] J. Kvarnström. (2012, April 7). *The UAV relay project*. [Online]. Available: http://www.ida.liu.se/divisions/aiics/aiicssite/projects/relay.en.shtml.

[20] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnström and P. Olsson, "Relay positioning for unmanned aerial vehicle surveillance," 2010 © The International Journal of Robotics Research. doi: 10.1177/0278364910369463.

[21] A. Holmberg and P. Olsson, "Route planning for relay UAV," in *26th International Congress of the Aeronautical Sciences*, 2008.

[22] Northrop Grumman. (2011). *Battlefield Airborne Communications Node and Global Hawk*. [Online]. Available: http://www.is.northropgrumman.com/products/bacn/assets/BACN.pdf.

[23] L. J. Austin III, "Joint publication 3–06. Joint urban operations," 2009.

[24] Cowling I.D., Whidborne J.F., Cooke A.K. and Yakimenko O.A., "Direct method based control system for an autonomous quadrotor," *J Intell Rob Syst Theor Appl Journal of Intelligent and Robotic Systems: Theory and Applications 60(2)*, pp. 285–316. 2010.

[25] Boeing. (2010, July 20). *ScanEagle UAV flight demonstrates narrowband communications relay*. [Online]. Available: http://boeing.mediaroom.com/index.php?s=43&item=1325.

[26] O. L. Gary, *Engineering Systems Integration: Theory, Metrics, and Methods.* Monterey, CA. CRC Press, 2012.

[27] A. Lacher and D. Maroney, “A new paradigm for small UAS,” 2012.

[28] J. Henderson. (2010, July). *Lethality Criteria for Debris Generated from Accidental Explosions*. [Online]. Available: http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA532158.

[29] Draganfly Innovation Inc. (2012, April 17). *Emergency measures*. [Online]. Available: http://www.draganfly.com/uav-helicopter/draganflyer-x6/applications/government.php#Emergency-Measures.

[30] L. Jones and O. Yakimenko. “Coordination and control for multi-quadrotor UAV missions,” M.S. thesis, Department of Mechanical Engineering, Naval Postgraduate School, Monterey, CA, 2012.

[31] Quanser. (2009). *Quanser Unmanned Vehicles Systems Laboratory Information Package.* [Online]. Available: http://www.quanser.com/english/downloads/products/UVS_Labs/Quanser_UVS_PIS.pdf.

[32] Quanser, “Quanser qball-X4 user manual,” unpublished.

[33] R. Murphy and S. Stover, “Gaps analysis for rescue robots,” *CRASAR-TR2005–12,* 2006.

[34] S. W. Paris and C. R. Hargraves, “OTIS 3.0 manual,” 1996.

[35] J. T. Betts and W. P. Huffman, “Sparse optimal control software SOCS,” *Mathematics and Engineer- ingAnalysis, TechnicalDocumentMEA-LR-085,* 1997.

[36] O. V. Stryk, “User’s guide for DIRCOL (Version 2.1): A direct collocation method for the numerical solution of optimal control problems,” Fachgebiet Simulation und Systemopti- mierung (SIM), 2000.

[37] I. M. Ross, “User’s manual for DIDO: a MATLAB application package for solving optimal control problems,” Tomlab Optimization Inc., TR 04–01.0, 2004.

[38] O. A. Yakimenko. “Simplified modification of the direct variational method for onboard solution of optimization boundary problems for flight vehicle trajectories,” *Journal of Computer and Systems Sciences International.* pp. 405. 1998.

[39] V. N. Dobrokhodov and O. A. Yakimenko. “Synthesis of trajectorial control algorithms at the stage of rendezvous of an airplane with a maneuvering object,” *Journal of Computer and Systems Sciences International C/C Of Tekhnicheskaia Kibernetika 38(2),* pp. 262–277, 1999.

[40] D. V. Alekhin and O. A. Yakimenko. “Control systems of moving objects - synthesis of optimization algorithm for planned-route flight trajectory by direct variational method,” *Journal of Computer and Systems Sciences International. 38(4),* pp. 650, 1999.

[41] O. A. Yakimenko. Class notes for ME4901 (direct methods for rapid prototyping of optimal maneuvers), Naval Postgraduate School.

[42] M. Fleiss, J. Levine, P. Martin and P. Rouchon, “Sur les systèmes non linéarités différentiellement plats,” in *C.R. Acad. Sci.*, Paris, vol.315, série I, pp. 619–624, 1992.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Director TDSI
   National University of Singapore
   Singapore

4. Senior Manager TDSI
   National University of Singapore
   Singapore

5. Mr Fong Saik Hay
   ST Engineering
   Singapore